



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## DETEKCE SVÁRU NA POLOTOVARU

TUBE WELDING DETECTION

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Lukáš Hvizdák

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Peter Honec, Ph.D.

BRNO 2018



# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Lukáš Hvizdák **ID:** 177508 **Ročník:** 3 **Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## Detekce svaru na polotovaru

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh řešení s vyšší přesností než současný systém detekce svaru na trubce hlavové opěrky pro automobily. Implementace do již fungujícího stoje sestávajícího s manipulátorem, lisu a kontrolní stanice.

1. Seznamte se s problematikou kamerových systémů a počítačového vidění.
2. Vyberte vhodnou line-scan kameru, objektiv a další komponenty vizuálního systému.
3. Vytvořte pracoviště pro otestování v laboratorních podmínkách.
4. Navrhněte a otestujte vhodnou metodu detekce pozice svaru.
5. Implementujte do Raspberry Pi nebo jiné vhodné platformy.
6. Navrhněte interface pro komunikaci s výrobní linkou.
7. Implementujte v průmyslovém provozu a otestujte.

### DOPORUČENÁ LITERATURA:

HLAVAC V., SONKA M., BOYLE R.: Image Processing, Analysis, and Machine Vision, ISBN 978-0495082521

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 21.5.2018

**Vedoucí práce:** Ing. Peter Honec, Ph.D.

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**

*předseda oborové  
rady*

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

# ABSTRAKT

HVIZDÁK, Lukáš. Detekce svaru na polotovaru. Bakalářská práce.

Cílem bakalářské práce je návrh řešení a výběr komponentů pro detekci svaru na polotovaru za účelem zvýšení spolehlivosti a přesnosti detekce než současný systém využívající dva optické jednobodové senzory intenzity odraženého světla. Jsou zde uvedeny komponenty pro systém detekce svaru. Dále je zde návrh testovacího pracoviště, sestavení testovacího pracoviště s implementací na jednodeskový počítač Raspberry Pi 3 B+, návrh metodiky detekce svaru a její implementaci do Raspberry pomocí programu napsaného v jazyce C++, popsání a sestavení výsledný vision systém. V závěru jsou diskutované výsledky práce.

## Klíčové slova

Bakalářská práce, počítačové vidění, detekce, Raspberry pi, řádková kamera, liniové světlo, zpracování obrazu.

# ABSTRACT

HVIZDÁK, Lukáš. Tube welding detection. Semester thesis.

The aim of the bachelor thesis is to design and select components for welding detection on the tube to increase reliability and accuracy of detection than the current two-point light intensity system. In bachelor thesis are components for the weld detection system. In addition, there is a design test site, a test site with a Raspberry Pi 3 B + implementation, a design of welding detection methodology and its implementation into Raspberry using a written C ++ program described and assembled resulting vision system. In conclusion, the results of the work are discussed.

## Keywords

Bachelor thesis, computer vision, detection, Raspberry pi, line camera, line light, image processing.

## **BIBLIOGRAFICKÁ CITACE**

HVIZDÁK, L. Detekce sváru na polotovaru. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 54 s. Vedoucí bakalářské práce Ing. Peter Honec, Ph.D..

## PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma „Detekce svaru na polotovaru“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 21.5.2018

.....  
podpis autora

Děkuji vedoucímu bakalářské práce Ing. Petru Honcovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce. Dále děkuji panu Ing. Karlu Flídrovi, za poskytnutí zázemí a možnost dělat bakalářskou práci ve společnosti Flídr s.r.o.

V Brně dne: **21 května 2018**

.....  
podpis autora

# Obsah

1	Úvod a cíl práce .....	11
1.1	Úvod.....	11
1.2	Cíl.....	11
1.3	Současné řešení detekce .....	12
2	Výběr kamery, objektivu a osvětlení .....	13
2.1	Měření svaru pomocí spektrometru .....	13
2.2	Výběr osvětlení .....	15
2.3	Výběr kamery – řádková kamera .....	16
2.3.1	Princip řádkové kamery .....	16
2.3.2	Výhody řádkové kamery oproti maticové .....	16
2.3.3	Basler racer .....	17
2.4	Objektiv.....	19
2.4.1	Nikon nikkor af 50mm f/1,8 .....	19
2.5	Raspberry Pi 3 .....	20
2.5.1	Raspberry Pi 3 B+ .....	20
2.5.2	100Mbit/s Ethernet .....	20
2.5.3	1000Mbit/s Ethernet .....	21
2.5.4	Napájecí zdroj .....	22
2.6	Gigabitový switch .....	22
3	Návrh pracoviště pro testování .....	23
3.1.1	Teoretický návrh testovacího pracoviště .....	23
3.1.2	Testovací pracoviště .....	24
4	Návrh vhodné metody detekce svaru.....	26
4.1	Jas.....	26
4.2	Kvantifikace .....	26
4.3	Kontrast .....	26
4.4	Ostrost .....	27
4.5	Řízení kamery .....	27
4.5.1	Binning – slučování sousedních pixelů .....	28
4.6	Způsob detekce svaru .....	28
4.6.1	Teoretický návrh .....	28
4.6.2	Implementace detekce do kódu.....	29
4.6.3	Výsledky reálné detekce svaru .....	31
5	Implementace do Raspberry pi 3 B+ .....	32
5.1	Výhody Raspberry Pi .....	32
5.2	Raspberry Pi a řádkové kamery .....	33
5.3	Popis kódu a funkcionality.....	34
6	Návrh interface pro komunikaci s výrobní linkou .....	38

6.1	Komunikační interface .....	38
7	Implementace do výroby .....	40
7.1	Výrobní linka .....	40
7.2	Návrh vision systému pro linku .....	41
8	Závěr .....	44
9	Citovaná literatura.....	46
	Seznam symbolů, veličin a zkratk.....	47
	Seznam příloh .....	48



# Seznam obrázků

Obr. 1-1: Trubka hlavové opěrky .....	12
Obr. 2-1: Měření svaru spektrometrem.....	13
Obr. 2-2: Intenzita ambientního světla .....	14
Obr. 2-3: Reflektance svaru vůči chromovému povrchu .....	14
Obr. 2-4: LND2-200RD .....	15
Obr. 2-5: Závislost intenzity osvětlení na vzdálenosti [7] .....	16
Obr. 2-6: řádková vs. maticová [1] .....	17
Obr. 2-7: Kamera Basler racer [2] .....	17
Obr. 2-8: Ohnisková vzdálenost objektivu [8].....	19
Obr. 3-1: návrh pracoviště pro testování .....	23
Obr. 3-2 Testovací pracoviště.....	24
Obr. 3-3 Držák kamery sestava.....	24
Obr. 3-4 Držák světla sestava .....	25
Obr. 4-1: Kontrast .....	26
Obr. 4-2: Binning [4] .....	28
Obr. 4-3: Ukázka kódu hledacího algoritmu .....	30
Obr. 4-4: Svar vůči normálnímu povrchu .....	30
Obr. 4-5: Průměrné hodnoty intenzity ze dvou měření .....	31
Obr. 5-1: Počet snímků zpracovaných za jednu sekundu .....	34
Obr. 5-2: Přirazení kamer nalezených na síti.....	35
Obr. 5-3: Nastavení kamer.....	35
Obr. 5-4: Zahájení snímání kamer .....	36
Obr. 5-5: Vyzvednutí obrazů od kamer .....	36
Obr. 5-6: Kontrola dat obrazu a vyhodnocení přítomnosti svaru .....	37
Obr. 6-1: GPIO sběrnice [7] .....	38
Obr. 6-2: Rozhraní pro operátora [10] .....	39
Obr. 7-1: Pohled na výrobní linku jako celek.....	40
Obr. 7-2: Zájmové části linky .....	41
Obr. 7-3: Návrh uchycení kamer a světel .....	42
Obr. 7-4: Boční pohled na sestavu kamer a světel.....	43
Obr. 9-1: Reflektance normálního povrchu vs. svaru .....	49
Obr. 9-2: Reflektance svaru pod červeným světlem.....	49
Obr. 9-3: Intenzita odrazu od normálního povrchu .....	50
Obr. 9-4: Intenzita odrazu od svaru .....	51
Obr. 9-5: Detail kamery, světla a motoru .....	54
Obr. 9-6: Pohled na svar testovací pracoviště.....	54
Obr. 9-7: OK kus s vyliisovanými drážkami .....	55
Obr. 9-8: Bod odkud se měří 500 mm .....	55

## Seznam tabulek

Tab. 2.3-1: základní specifikace raL2048-48gm .....	18
Tab. 2.4-1 Základní specifikace objektivu.....	19
Tab. 2.5-1: Parametry zdroje DR-60-5 .....	22
Tab. 6.1-1: Komunikace s linkou.....	38
Tab. 7.2-1: Vybrané hodnoty z měření času trvání cyklu.....	52
Tab. 7.2-2: Tabulka středních hodnot obrazu .....	53

# 1 ÚVOD A CÍL PRÁCE

## 1.1 Úvod

Počítačové vidění je relativně nový a dynamický obor v měřicí a výpočetní technice. K rozmachu došlo hlavně v posledním desetiletí díky nárůstu výkonu výpočetní techniky, ale zájem je značný už poslední tři dekády. Hlavně v aplikacích jako je vzdálené snímání a měření, autonomní řízení vozidel, biomedicinské zobrazování a automatizovaný dohled. Rovněž kamerové systémy ušly značnou cestu a výrazně zlevnily, k čemuž dopomohlo i používání digitálních kamer a fotoaparátů běžnými lidmi v každodenních situacích. Dalším důležitým faktorem je lepší a větší dostupnost různých materiálů, návodů a algoritmů, zaměřujících se právě na zpracování obrazu, a to jak komunitami lidí, kteří své zkušenosti a poznatky sdílí na internetu, tak i výrobci samotných kamerových systémů a v neposlední řadě vysokými školami.

V každodenním životě se s počítačovým viděním setkáváme už teď v podstatě na denní bázi. Například průjezdové radary na silnicích, nebo rozpoznávání překážek, jako jsou chodci a zvěř, které se řadí do příplatkové výbavy automobilů, dále přizpůsobení podvozku k technickému stavu komunikace, po které se vozidlo pohybuje, ale také kontrola stavu kotevních lan, potrubí a dalších průmyslových zařízení pomocí dronů. Toto jenom malý výčet aplikací počítačového vidění, sloužící pouze pro ilustraci rozmanitosti využití. [1]

## 1.2 Cíl

Cílem bakalářské práce je návrh řešení metody detekce, výběr komponentů pro detekci svaru a realizace systému pro detekci svaru na polotovaru (trubce, obrázek 1.2-1) za účelem zvýšení spolehlivosti a přesnosti detekce, než má současný systém využívající tři bodové senzory intenzity odraženého světla. Trubka je součástí hlavové opěrky do aut. Jelikož se jedná o jeden z bezpečnostních prvků každého vozu, musí podléhat přísným crash testům. Svar je nutné detekovat především kvůli lisování drážek pro uchycení opěrky do autosedačky. Pokud by došlo ke špatnému natočení, je možnost, že se drážky vylišují přes svar anebo jeho okolí, což by vedlo k vytvoření prasklin, které by oslabily celou konstrukci opěrky. Takováto závada by měla za následek zranění při autonehodě, pokud by se dostala do oběhu. Vadné kusy se objevují s cca 5 % výskytem v sérii o několika tisíci kusech. Pokud se podaří toto číslo zmenšit, nejenže to ušetří peníze společnosti Flídr s.r.o., pro kterou tuto práci vyhotovuji ale také to uspokojí auditory z automobilek, pro něž se tyto opěrky vyrábí.



**Obr. 1-1: Trubka hlavové opěrky**

### **1.3 Současné řešení detekce**

V současnosti společnost Flídr s.r.o. používá několik způsobů detekce svaru. Jsou implementovány na různých linkách, každá linka pro specifickou hlavovou opěrku do různých automobilů. Minoritně zastoupeny jsou metody mechanické detekce a detekce pomocí vířivých proudů. Mechanická detekce funguje na principu přiloženého trnu, který sepne spínač při najetí na svar. Svar má tendenci být vystouplý o 0,1 mm až 0,2 mm na vnitřní straně trubky. Bohužel se vyskytují i polotovary s menším výstupkem, nebo naopak s propadlinou v místě svaru.

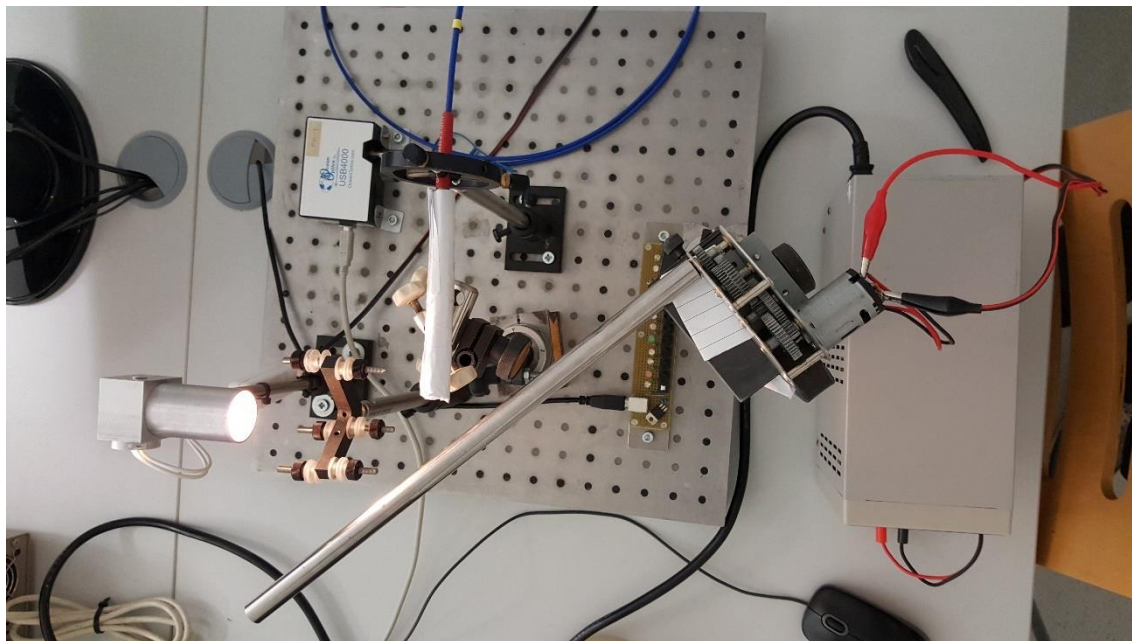
Detekce pomocí vířivých proudů zajišťuje programovatelná sonda s digitálním výstupem, která je přiložena k povrchu trubky. Vzhledem k malé tloušťce pláště polotovaru je tato metoda nepřesná, jelikož jsou zde jenom malé rozdíly v proudech v okolí svaru, vůči zbytku povrchu. Jedná se pouze o jednu implementaci.

Nejvíce zastoupená je detekce pomocí odrazu světla. Prakticky skoro u všech linek jsou různé senzory intenzity s vlastní osvětlovací diodou v těle snímače. Hlavní nevýhody toho řešení představují: velká snímaná plocha v řádech od 1 mm do 3 mm, vzhledem k rozměru polotovaru pouze lokální bodové měření a v poslední řadě nižší rychlost měření odvíjející se od typu daného senzoru. V současné době je snaha tyto nevýhody minimalizovat použitím více senzorů s rozestupy na jeden polotovar, typicky dva senzory až tři. Pokud ovšem na polotovaru je znečištění od chladicí emulze nebo jiný druh nečistoty, často dochází ke špatnému vyhodnocení a záměně za svar. Emulze se může vyskytovat na polotovaru z důvodu řezání na potřebné rozměry z pětimetrových trubek přímo ve výrobě.

## 2 VÝBĚR KAMERY, OBJEKTIVU A OSVĚTLENÍ

### 2.1 Měření svaru pomocí spektrometru

K zjištění nejvhodnějšího spektra světla, ve kterém je největší rozdíl mezi normálním povrchem trubky (chromově lesklý) a svarem (tmavě modrý až fialová) jsem použil spektrometr OceanOptis USB4000.

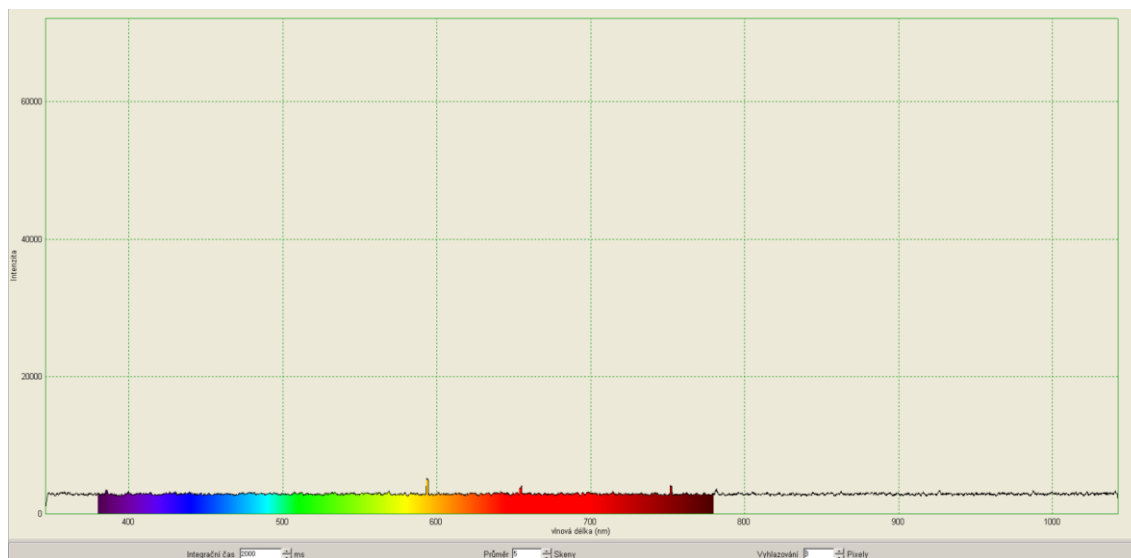


Obr. 2-1: Měření svaru spektrometrem

Provedl jsem několik měření, která zahrnovala různý úhel dopadu a odrazu světla.

Hlavní však bylo měření se stejným úhlem dopadu jako odrazu, kvůli povrchu trubky, který reflektuje většinu dopadajícího světla především přímým odrazu. Odraz difuzního světla je téměř zanedbatelný a za běžných podmínek pohlcen okolním ambientním světlem. Ambientní světlo, které jsem nemohl nijak ovlivnit, pocházelo z monitoru počítače a zbytek přirozeného světla prostupovalo skrze okenní žaluzie. Toto světlo bylo odfiltrováno programem pomocí kalibrace. Intenzita tohoto světla je níže na obrázku 2.2.

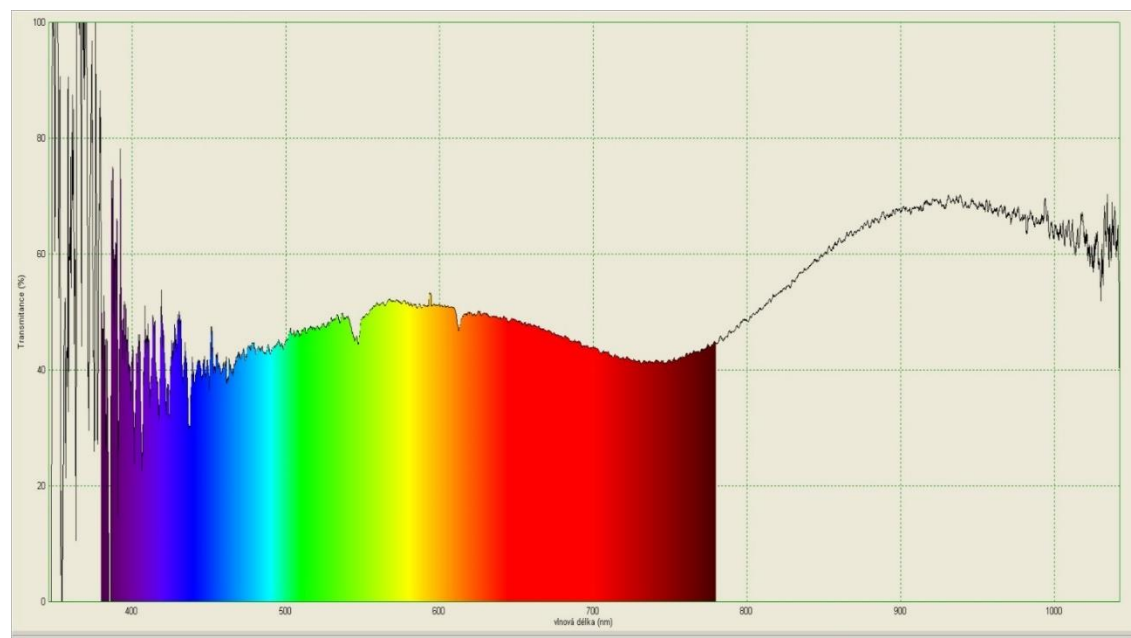
Za provozu bude ve výrobě ambientní světlo přesvícené liniovým osvětlením na lince. Ambientní světlo na pracovišti je kvůli druhu výroby až 750 luxů. Osvětlení na pracovišti se řídí normou ČSN 730580-4 - denní osvětlení průmyslových budov 300 – 500 luxů a všeobecnou normou Osvětlení všeobecně ČSN 12 464-1. Intenzita liniového světla a jeho problematika je uvedena níže v kapitole 2.2 Výběr osvětlení.



**Obr. 2-2: Intenzita ambientního světla**

Spektrum zdroje světla lze nalézt v příloze 1.

Na obrázku 2.3 je reflektance<sup>1</sup> světla dopadajícího a odrážejícího se pod úhlem 45° od svaru na povrchu trubky při zapnutém zdroji světla. Spektrometr jsem kalibroval před měřením na odraz od chromu při zapnutém zdroji světla i při vypnutém. Tím jsem byl schopný změřit jinou intenzitu odraženého světla od svaru vs. Normálního povrchu. Měření s jiným úhlem odrazu než dopadu (měření difuzního světla) jsem provedl rovněž, ale výsledný graf vyšel stejný, pouze se lišil podstatně menší intenzitou odraženého světla.



**Obr. 2-3: Reflektance svaru vůči chromovému povrchu**

<sup>1</sup> Reflektance – je množství světla, které se odrazilo od vzorku. Vždy se měří vůči blanku, který má reflektanci 100 % a reflektance vzorku je vždy menší než 100 %.

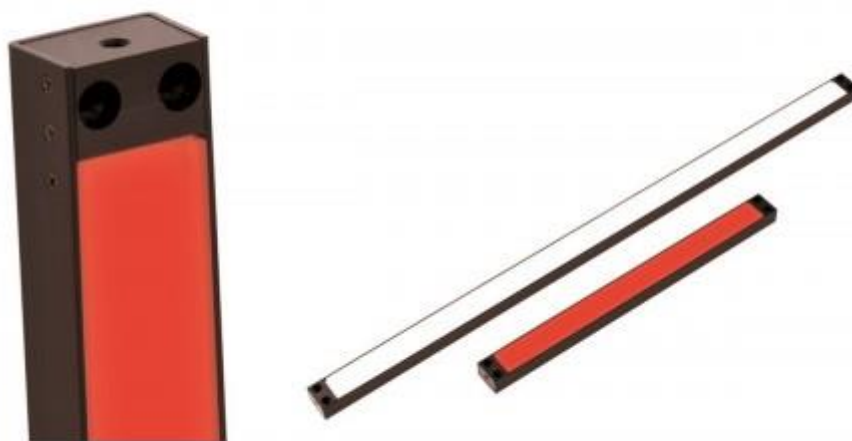
Další grafy v plném rozlišení jsou přiložené v příloze 1 (Měření trubky spektrometrem).

Z měření vyplynulo, že v celém viditelném spektru (400–780 nm) pohlcuje svar kolem 50 % dopadajícího světla. Nejvíce svar pohlcuje světlo v 450 nm (modrá) a v 760 nm (červená). Vzhledem k tomuto faktu na vlnové délce osvětlení tolik nezáleží, protože 50 % úbytek je spolehlivě detekovatelný.

## 2.2 Výběr osvětlení

Světla pro práci s řádkovými kamerami musí splňovat několik základních požadavků. Hlavní parametr při výběru osvětlení je intenzita světla, jelikož řádkové kamery mají extrémně krátký expoziční čas v řádech mikrosekund. Vysoké intenzity se dosahuje zpravidla usměrňovací čočkou nebo zrcadlem. Další důležitý aspekt je rovnoměrnost osvětlení a stálost výkonu i frekvence v čase. U monochromatických kamer se doporučuje použití monochromatického světla (červená, zelená, ...), spíše než bílého, aby se eliminoval vliv barevné vady optiky. [2]

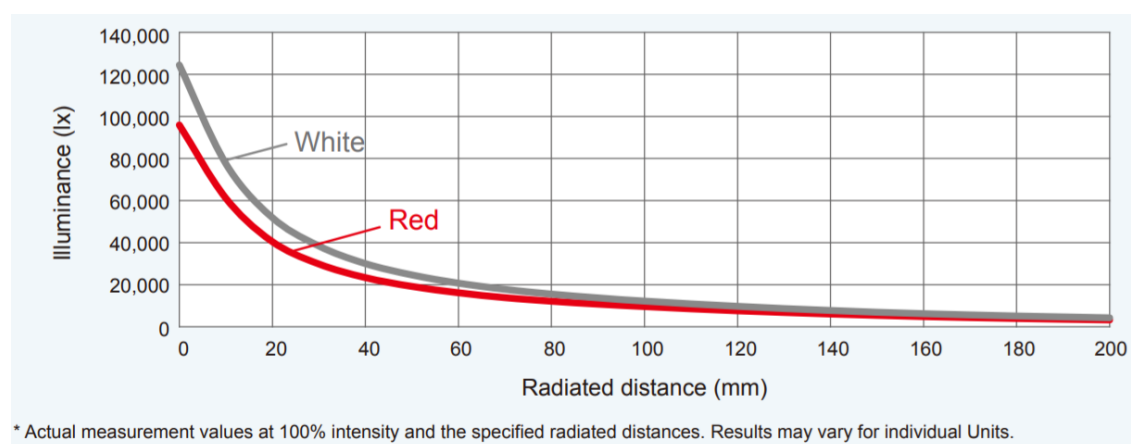
Jako nejvhodnější se jeví použití liniového světla s difuzní a zaostřovací čočkou od výrobce CCS řada LND2-200RD s červeným světlem (630 nm) emitovaným led diodami. Čip kamery je nejefektivnější mezi 600 nm až 700 nm vlnové délky světla, viz. katalog kamery v příloze 4.



Obr. 2-4: LND2-200RD

Světlo je napájeno DC 24 V a výkon činný 22 W. Červenou barvu jsem oproti jiným barvám upřednostnil kvůli větší dostupnosti a zároveň použití monochromatického osvětlení nám pomůže odfiltrovat ambientní světlo, které by mohlo působit rušení.

Rovněž je méně rušivé pro své okolí, kvůli odlišné spektrální citlivosti lidského oka. Na obrázku 2.5. je závislost intenzity osvětlení [lx] na vzdálenosti od objektu v [mm]. [2] K napájení světla použiji zdroj MEAN WELL DRA-60-24 o maximálním výkonu 60 W s PWM regulací výstupu 10 až 100 %. [3] Katalog je v příloze 2 (Světlo).



Obr. 2-5: Závislost intenzity osvětlení na vzdálenosti [7]

## 2.3 Výběr kamery – řádková kamera

### 2.3.1 Princip řádkové kamery

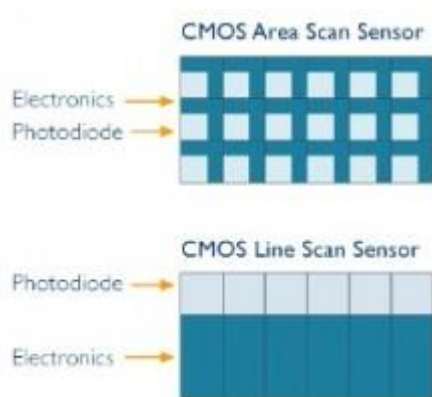
Snímač u řádkové kamery je většinou tvořen jedinou linií obrazových bodů (pixelů). Kamera pracuje v podstatě stejně jako skener v kopírce. Kamera skládá obrázek z jednotlivých řádků, jelikož je kamera nehybná, tak je nutné, aby se sledovaný objekt pohyboval. Po nasnímaní určitého počtu dat pošle tzv. frame<sup>2</sup> do nadřazeného vision systému, většinou tvořeného počítačem, který jej dále zpracovává. Tento způsob snímání umožňuje snímat objekty pohybující se velkou rychlostí s velkým detailem. [2]

### 2.3.2 Výhody řádkové kamery oproti maticové

Při snímání objektu s vysokou rychlostí klasickou kamerou s maticovým snímačem je nutné mít silné zábleskové světlo a velmi krátký expoziční čas, jinak dojde k rozmazání objektu jeho vlastním pohybem, zatímco lineární snímač míří přímo na oblast zájmu a nepřenáší se tak nedůležité části obrazu. To mu umožňuje snímat s velmi krátkým expozičním časem a je schopný nasnímat až několik desítek tisíc řádků za sekundu. K nasvícení nám postačí liniové světlo, které je schopné soustředit veškerý světelný tok

<sup>2</sup> Frame – obrázek, rámec, matice o definovatelném počtu řádků.





**Obr. 2-6: řádková vs. maticová [1]**

U maticového senzoru je část plochy pixelu vyhrazena pro elektroniku, kdežto u lineárního senzoru je celá plocha pixelu vyhrazena pro světlo citlivou buňku. Elektronika, která zpracovává signál ze světlo citlivé buňky, je umístěná bokem. Díky tomuto uspořádání má řádková kamera až dvakrát větší světelnou citlivost. [2]

### 2.3.3 Basler racer

Pro danou aplikaci jsem zvolil monochromatickou kameru Basler racer. Je to typickým zástupcem jednoduché lineární kamery, jejíž senzor tvoří jediný řádek pixelů. Díky technologií CMOS s vysokou kvantovou účinností je u senzoru dosažena co nejlepší citlivosti. Jednotlivé obrazové body jsou na senzoru uspořádaný tak, aby využily co nejefektivněji celou plochu senzoru. Kamera dle jednotlivých typů má rozlišení od 2 tisíc do 12 tisíc pixelů. S využitím komunikačního rozhraní Camera link a technologie řádkové kamery jsme schopni dosáhnout frekvenci snímání až 80kHz. Výhodou kamery je jednoduchá konstrukce, snadné nastavení a nízká cena. [2]



**Obr. 2-7: Kamera Basler racer [2]**

Řada racer má celkem pět typů kamer z toho každý typ se dělí ještě na dva poddruhy podle použitého komunikačního rozhraní, a to Gigaethernet nebo zmíněný Camera link. Jednotlivé typy se liší velikostí rozlišení, a tím i délkou senzoru a velikostí jednotlivých pixelů. S tím souvisí i maximální frekvence, kterou je daná kamera schopna nabídnout s rozhraním Gigaethernet. U rozhraní Camera link je frekvence snímání shodně 80 kHz pro všechny kamery, kromě varianty 12k, která zvládne 66 kHz. Pro moji aplikaci se jako nejvhodnější jeví racer raL2048-48gm.

rozlišení	2048
typ senzoru	Awaiba DR-2k-7 Monochrome Linear CMOS
rozměry pixelu	7 μm x 7 μm
bitová hloubka	až 12 bit
délka senzoru	14,34 mm

**Tab. 2.3-1: základní specifikace raL2048-48gm**

Výpočet minimálních snímku za vteřinu, pokud chci narazit v jedné otáčce alespoň jednou na svar.

Rychlost otáčení polotovaru je  $360^\circ \cdot s^{-1}$ .

Svar zabírá  $25^\circ$ .

Vzhledem k tomu, že se neví, v jaké pozici je svar při začátku hledání tak: se polotovar musí otočit alespoň o  $360^\circ + 25^\circ$ , s rezervou  $400^\circ$ . Hodnota minimálního kroku vzorkování vzhledem k svaru by měla být alespoň  $1/10$  svaru, tj.  $2,5^\circ$ .

Výpočet tedy bude:

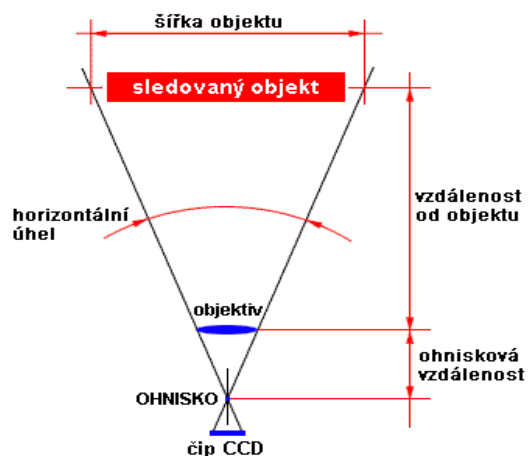
$$\frac{\text{Počet stupňů za sekundu}}{\text{minimální krok}} = \frac{360}{2,5} = 144 \text{ FPS} \quad (1)$$

Více o kameře je možné dohledat v katalogu [4], který je přiložen v příloze 4 (BaslerRacer).

Napájecí zdroj použiji 12 V, HRS 12 pin, od výrobce Basler určený přímo pro napájení kamer Basler racer.

## 2.4 Objektiv

Na objektivu jsou kladeny specifické požadavky podle využití. U objektivu pro řádkové kamery tomu není jinak. Díky standartní velikosti pixelů není kladen důraz na rozlišení objektivu, jako na schopnost vykreslit obraz i na senzory o velikosti až několik centimetrů



ale s co nejmenším vlivem vinětace<sup>3</sup>. Proto je kladem obrovský důraz na rozložení jasu a kontrastu na celém zorném poli. Ty nejlepší objektivy jsou vyráběny pro specifický rozsah zvětšení, ve kterém je garantován stálý optický výkon. [2] Pro účely semestrální a bakalářské práce není zapotřebí tak přísných kritérií, jelikož nebudu skládat obraz ale pouze měřit množství odraženého světla.

Obr. 2-8: Ohnisková vzdálenost objektivu [8]

### 2.4.1 Nikon nikkor af 50mm f/1,8

Na doporučení pana Ing. Petra Honce, Ph.D. jsem zvolil právě tento objektiv.

Ohnisková vzdálenost	50 mm
Světelnost	f/1,8
Obrazový úhel	46°
Nejmenší zaostřovací vzdálenost	450 mm

Tab. 2.4-1 Základní specifikace objektivu

Výpočet vzdálenosti, která je nutná pro dosažení požadovaného záběru kamery ral2048 14.43 mm senzor a s daným objektivem. Viz. obrázek 2.4-1.

Focal length<sup>4</sup> -  $f = 50 \text{ mm}$

Délka senzoru -  $d_s = 14,43 \text{ mm}$

Délka detekce na trubce –  $d_T = 150 \text{ mm}$

Požadovaný vzdálenost od trubky –  $b = ?$

$$\frac{b}{d_T} = \frac{f}{d_s} \rightarrow \frac{b}{150} = \frac{50}{14,43} \rightarrow b = \frac{50}{14,43} \cdot 200 = 519 \text{ mm} \quad (2)$$

<sup>3</sup> Vinětace – vada optických soustav, projevující se nižším jasem na okrajích zobrazovaného obrazu.

<sup>4</sup> Focal length – Ohnisková vzdálenost

Vzdálenost 519 mm je dostatečná i pro zaostření objektiv. Podobným způsobem jsem vypočetl výšku výseče snímané plochy, která činí 70  $\mu\text{m}$ .

## 2.5 Raspberry Pi 3

Raspberry Pi 3 je jednodeskový počítač o rozměrech kreditní karty vyvinutý britskou nadací Raspberry Pi Foundation. Vybral jsem jej kvůli přímé podpoře společnosti Basler pro jejich kamery. Díky malým rozměrům je velice kompaktní a GPIO porty umožňují spínání relátek. Výkon zajišťuje 64 bitový, čtyř jádrový procesor od společnosti ARM s taktem jednoho jádra 1,2 GHz.

Operační systém Raspbian zajišťuje plnou podporu jak pro Pylon 5<sup>5</sup> tak samozřejmě podporu GPIO portů a všechny důležité drivery.

### 2.5.1 Raspberry Pi 3 B+

V průběhu řešení bakalářské práce vyšla vylepšená verze jednodeskového počítače Raspberry Pi 3 B označována jako Raspberry Pi 3 B+.

Součástí této novinky je nový síťový čip, který umožňuje komunikaci rychlosti 1000Mbit/s. Tím se snižuje vliv nízké přenosové rychlosti původního Raspberry Pi 3 B, které působilo jako úzké hrdlo. Vzhledem k tomu, že pro aplikaci je třeba dvou kamer s 1000Mbit/s, tak pořad nebudou kamery schopny dosáhnout maximální možné přenosové rychlosti, ale to již není zásadní ani omezující faktor.

Dalším minoritním zlepšením nového modelu je navýšení rychlosti procesu z 1,2 GHz na 1,4 GHz. Zapouzdření procesu je oproti předchozí generaci kovové, což umožňuje lepší odvod tepla z povrchu procesoru.

### 2.5.2 100Mbit/s Ethernet

Výpočet velikosti jednoho snímku přeneseného 100Mbit/s Ethernetem. [4]

Největší doporučená velikost packetu pro 100Mbit/s Ethernet je 1500 bytů při prodlevě mezi packety 11772 tiků. [5] Podíl payload size<sup>6</sup> s packet size<sup>7</sup> se zaokrouhluje na celé číslo a payload size na násobky čísla 4. [4]

---

<sup>5</sup> Pylon 5<sup>5</sup> - software pro nastavení kamery, ovladačů a obsahuje vývojářský kit

<sup>6</sup> Payload size – velikost snímku v bytech vypočtena z rozlišení a bitové hloubky

<sup>7</sup> Packet size – velikost jednoho packetu skládajícího se z payload a řídicích bitů

Pro výpočet jsem předpokládal bitovou hloubku černobílého obrazu 8 bitů

$$\frac{Bytes}{Frame} = \left[ \frac{Payload\ size}{Packet\ size} \cdot Packet\ Overhead \right] + Payload\ size + Leader\ Size + Trailer\ Size \quad (3)$$

Kde:

Packet Overhead<sup>8</sup> = 72 (for a GigE network) 78 (for a 100 MBit/s network)

Leader Size<sup>9</sup> = Packet Overhead + 36 (if chunk mode is not active)

Packet Overhead + 12 (if chunk mode is active)

Trailer Size<sup>10</sup> = Packet Overhead + 8 [4]

Dále pro kameru Basler racer raL2048-48gm jeden snímek má velikost:

$$\left[ \frac{2048}{1500} \cdot 78 \right] + 2048 + 114 + 86 = 2354\ Bytes \quad (2.1)$$

Z výpočtů vyplývá, že dvojce kamer přes 100Mbit/s ethernet jsou schopny přenést každá 2655 snímků za sekundu. Vzhledem k tomu, že je nutná i zpětná komunikace s kamerami, je toto číslo nedosažitelné a pro danou aplikaci zbytečně velké.

Dokumenty [5], [4] jsou v příloze 3 (RaspberryPi).

### 2.5.3 1000Mbit/s Ethernet

Kontrolní propočet, možnosti přenosu jednoho snímku o rozměrech 2048 x 1 pixlů přes 1000Mbit/s ethernet. Výpočet vychází z předchozí pod kapitoly 2.5.2. Proto jej zde nebudu podrobněji rozebírat.

Velikost jednoho snímku poslaného přes 1000Mbit/s.

$$\left[ \frac{2048}{1500} \cdot 72 \right] + 2048 + 108 + 80 = 2334\ Bytes \quad (3.1)$$

---

<sup>8</sup> Packet Overhead - Čas potřebný k přenosu dat v síti s přepojováním paketů

<sup>9</sup> Leader Size – velikost inicializačních dat přenosu

<sup>10</sup> Trailer Size – velikost ukončovacích dat přenosu

Výpočet počtu snímku teoreticky přenesených při jedné kameře za jednu vteřinu.

$$\frac{\text{Počet bytů za vteřinu}}{\text{Velikost jednoho snímku}} = \text{Teoretický FPS}^{11} \quad (4)$$

$$\frac{125\,000\,000}{2\,334} = 53\,556 \text{ FPS} \quad (4.1)$$

Pro dvě kamery zapojené přes switch se dá předpokládat pokles alespoň na polovinu snímků za vteřinu přenesených po 1000 Mbit/s ve výpočtu 2.4. To znamená 26 778 snímků za vteřinu z každé kamery. Kamery data taky přijímají, a zároveň není do objemu dat po lince přičtený komunikační protokol ani jeho charakteristiky, proto je tato rychlost pouze teoretická ale i tak dalece přesahuje požadavky pro aplikaci. Výpočet dostatečných FPS je v přechozí kapitole 2.3.3 Basler racer.

## 2.5.4 Napájecí zdroj

Zdroj Mean Well DR-60-5 je spínaný zdroj určený do průmyslu pro montáž na DIN lištu. Bude sloužit jako napájení raspberry pi 3 a také i gigabitového switche. Katalog je v příloze 3 (RaspberryPi).

Výstup		Vstup	
napětí U [V]	5	Počet fází	1
proud I [A]	6,5	Odběr (230 VAC)	0,8 [A]
výkon P [W]	32,5	Účinnost	76 %
max. zvlnění [mV]	80		
konst. napěti	Ano		

Tab. 2.5-1: Parametry zdroje DR-60-5

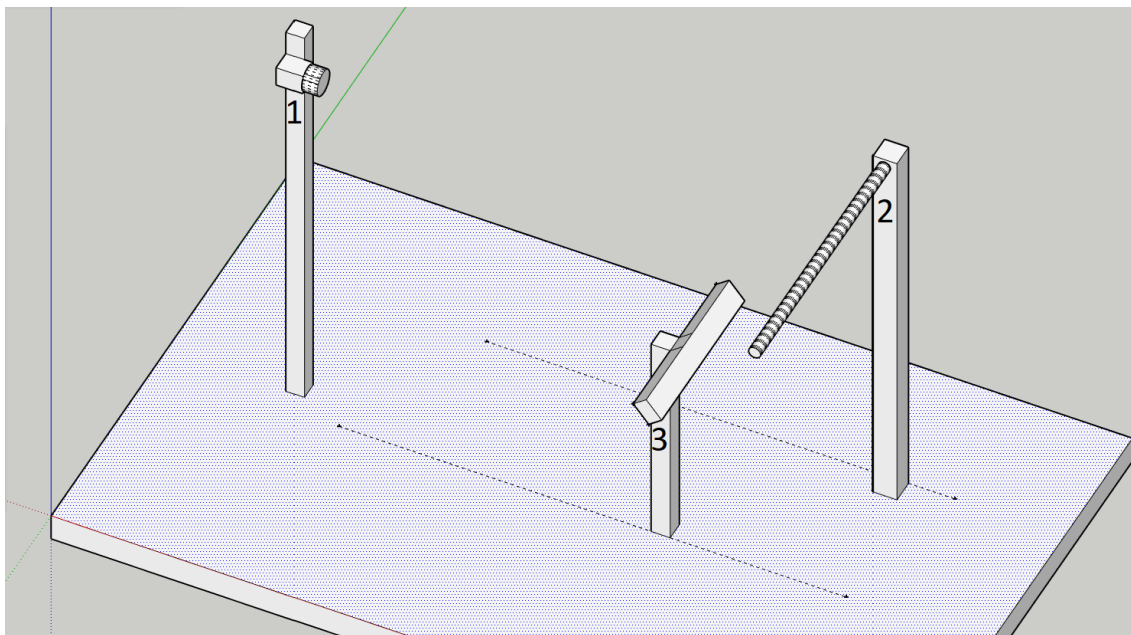
## 2.6 Gigabitový switch

Vybral jsem gigabitový switch D-Link DGS-105 s 5 o propustnosti 10Gbps. Podporované jsou dále standartní protokoly 802.3 (10BASE-T), 802.3u (100BASE-TX), 802.3ab (1000BASE-T Gigabit Ethernet) a 802.3x nebo 802.1p Qos.

Napájený je 5 V o odběru 1 A.

<sup>11</sup> FPS – Frame Per Second – snímky za vteřinu

### 3 NÁVRH PRACOVISTĚ PRO TESTOVÁNÍ



Obr. 3-1: návrh pracoviště pro testování

#### 3.1.1 Teoretický návrh testovacího pracoviště

Na obrázku 3.1 je hrubý návrh, jak bych chtěl sestavit testovací pracoviště. Číslo 1 označuje kameru, číslo 2 je trubka, číslo 3 liniové světlo.

Uchycení kamery bude mít možnost výškového posunu, a i natočení pro změnu úhlu kamery, a tím i jejího zorného pole.

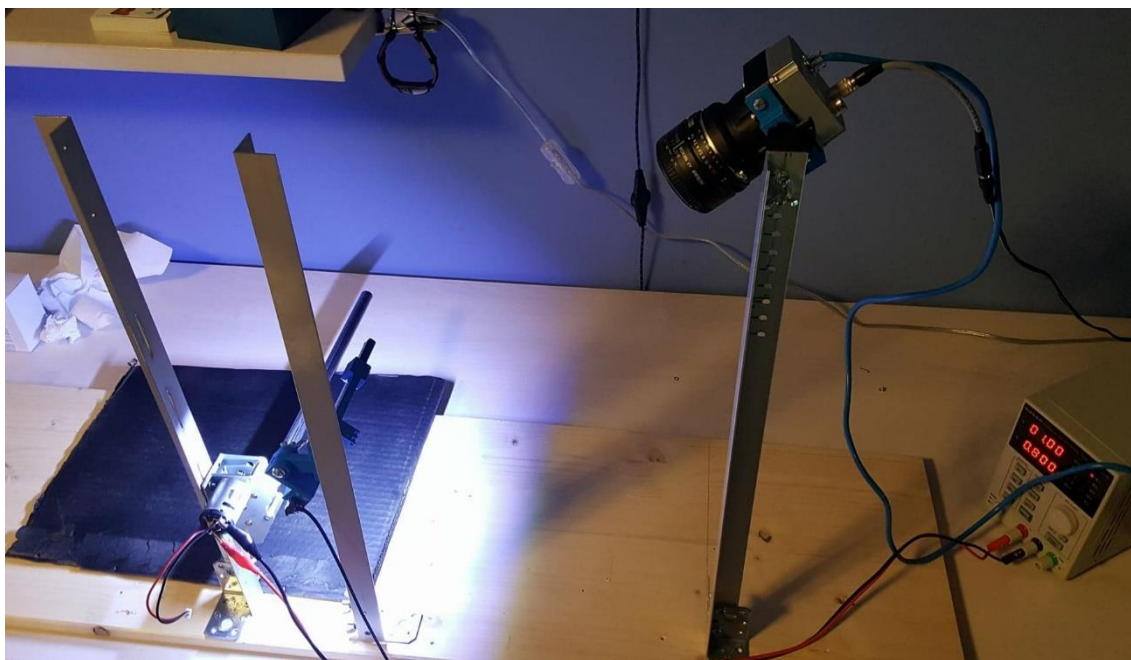
Trubka se bude otáčet na stejnosměrném motoru a držák bude rovněž výškově nastavitelný. Zároveň se bude dát měnit vzdálenost od kamery přemístěním celého stojanu.

Stojan světla bude rovněž pohyblivý pro změnu vzdálenosti od trubky, výškově nastavitelný a se změnou úhlu natočení světla.

Elektronika ve smyslu napájecích zdrojů bude bokem. K napájení použiji buď zdroje uvedené v kapitole dva nebo zapůjčené laboratorní zdroje ze školy.

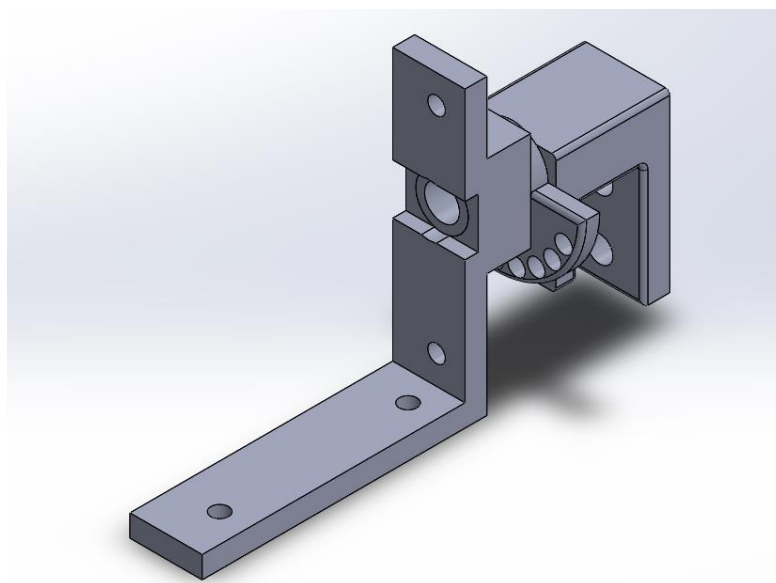
Testovací pracoviště bude sloužit především pro vývoj software a ověřování funkčnosti algoritmu detekce svaru.

### 3.1.2 Testovací pracoviště



Obr. 3-2 Testovací pracoviště

Testovací pracoviště se skládá z držáků kamery, držáků světla a stejnosměrného motoru otáčejícím se vzorkem polotovaru. Držák kamery je umístěný na hliníkovém L profilu s vyvrtanými dírami pro možnost výškového posunu. Hliníkový L profil je připevněn vruty k desce. Samotný držák kamery jsem vyprojektoval v program Solid Works a byl vytištěný na 3D tiskárně.

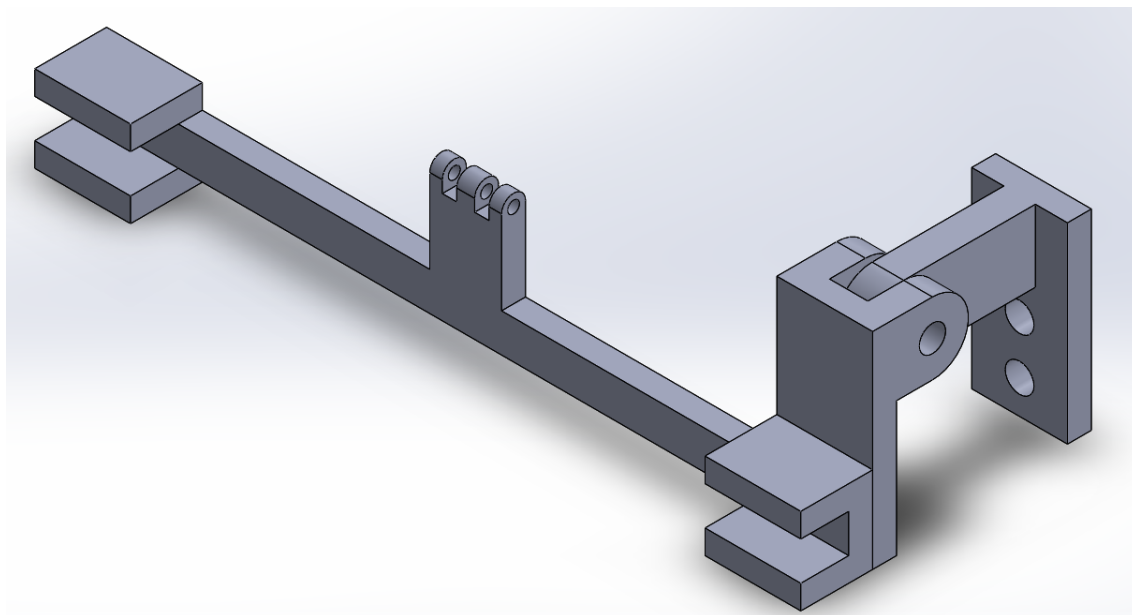


Obr. 3-3 Držák kamery sestava



Samotný držák kamery sestává ze tří dílů, jeden díl slouží k uchycení na L hliníkový profil, další dva díly drží samotnou kameru a fixují natočení. Původně se jednalo o jeden jednolitý díl, ale kvůli 3D tisku se musel rozdělit na dva samostatné díly, které do sebe zapadnou. Do středového otvoru patří šroub velikosti M6 sloužící jako středová osa.

Dalším dílem tištěným na 3D tiskárně byl držák světla obr. 3-4, který se skládá ze tří samostatných kusů. Uchycení držáku bylo dvakrát změněno kvůli špatnému odhadu vůlí a pevnosti drobných částí vytištěných z PLA materiálu na 3D tiskárně. Původně byl kloub umožňující natočení umístěn uprostřed. V pozdějším návrhu jsem jej přesunul na kraj a byl přilepen lepidle. Samotný L profil místo děr má v sobě drážky o šířce cca 5.5 mm zajišťující provlečení šroubů velikosti M5 a jejich aretaci pomocí matky.



**Obr. 3-4 Držák světla sestava**

Světlo použité pro testování není určeno pro tuto aplikaci. Jedná se o světlo pro denní svícení do automobilu. Jeho výkon je čtvrtinový oproti plánovanému osvětlení do výroby. Vzhledem k tomu, že se jednalo o přímou reflexi, tak i během měření a testování za slunečního dne nedocházelo k neschopnosti rozeznat svar v programu.

Reálné provedení je v příloze 6 (Testovací pracoviště).

K otáčení trubky sloužil DC motor s převodkou. Převodovka zajišťovala nízké otáčky. Celý klompet jsem koupil v modelářském obchodě, jedná se původně o ústrojí do modelu tanku.

Výsledky testování jsou komentovány v kapitole 4 Návrh vhodné metody detekce svaru. Testování proběhlo jak na notebooku, tak i přímo na Raspberry PI.

## 4 NÁVRH VHODNÉ METODY DETEKCE SVARU

### 4.1 Jas

U monochromatické (černo-bílé) kamery je nejdůležitějším parametrem jas, který také může vyjadřovat ostatní fyzikální veličiny (teplotu, rozložení tlaku, vzdálenost od pozorovatele atd.). Jelikož jas vyjadřuje různé veličiny, jeho použitím jako základní veličiny se vyhneme komplikovaným matematickým výpočtům za účelem vytvoření obrazu. [1]

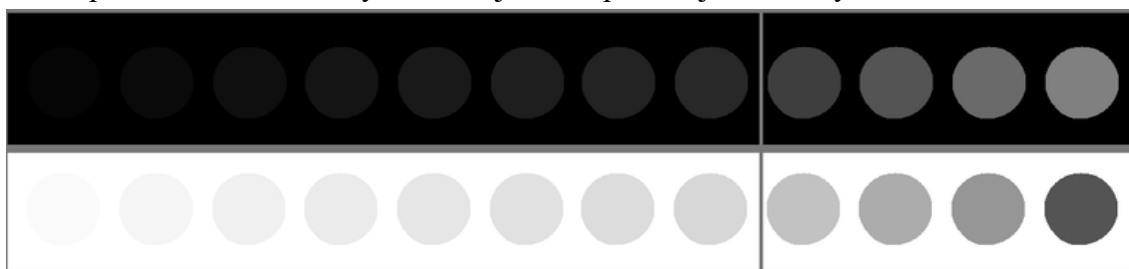
### 4.2 Kvantifikace

Převod mezi kontinuálními hodnotami funkce obrazu (jas) a jejími digitálními ekvivalenty se nazývá kvantování. Počet kvantizačních úrovní by měl být dostatečně vysoký, aby umožnil lidskému vnímání rozeznat jemné stínování a detaily v obraze. Většina zařízení pro zpracování digitálního obrazu používá kvantování o  $k$  stejných intervalech. Bity  $b$  jsou použity pro vyjádření hodnot jasu pixelů. Potom počet úrovní jasu se vypočte jako  $k = 2^b$ . Běžné je 8 bitů na jednu barvu ale můžeme se setkat i s úrovněmi (4,6,12...).

Efektivní výpočet hodnot jasu v digitálním obraze vyžaduje osm bitů, což odpovídá jednomu bytu. [1]

### 4.3 Kontrast

Kontrast je lokální změna jasu a je definovaný jako poměr mezi průměrným jasem a pozadím objektu. V podstatě bychom měli mluvit o svítivosti<sup>12</sup> místo o jas, pokud se chceme držet fyzikální přesnosti. Lidské oko je logaritmičtě citlivé na jas, což znamená, že pro vnímání vyššího jasu potřebujeme vyšší kontrast. [1]



Obr. 4-1: Kontrast

<sup>12</sup> Svítivost – popisuje množství světla které projde skrz nebo je vyzařované z bodové oblasti v různých směrech.

## 4.4 Ostrost

Ostrost je schopnost detekovat detaily v obraze. Lidské oko je méně citlivé na pomalé či rychlé změny v jas, ale je mnohem citlivější na rozsáhle změny v obraze jako je například pohyb. Ostrost rovněž klesá se zvětšující se vzdálenosti od optické osy. [1]

Zaostření se v průběhu testování ukázalo jako důležité, jelikož nezaostřený obraz byl lehce ovlivňován okolním ambientním a difuzním světlem. I přes to bylo možné svar nalézt a vnášelo to do procesu zbytečnou chybovost.

## 4.5 Řízení kamery

Téměř všechny řádkové kamery podporují nastavení parametrů jako frekvence snímání, délka expozic, nastavení režimu spouštění a další. Řádková kamera posílá takzvaný frame<sup>13</sup>, který se skládá z předem definovaného počtu řádků. Zatímco maximální velikost frame je definovaná, minimální nikoliv. Proto lze poslat i jeden nasnímaný řádek i když se tak zmenšuje efektivita přenosu. Režimy spuštění kamery jsou následující: [6]

**Volnoběžný režim:** Předem je nastavený počet řádků obrazu a frekvence snímání. Tento režim je vhodný pouze, když jsme schopni zajistit stálou rychlost snímaného objektu. Proměnná rychlost by měla za následek deformaci obrazu. [6]

**Externí synchronizace:** Pomocí digitálních vstupů kamery, na které se připojí digitální čidlo nebo kvadrurní enkodér, jsme schopni spouštět expozice jednotlivých řádků. Díky tomu je tento režim vhodný, když kolísá rychlost snímaného předmětu. Mimo vstupy ovládající spuštění jednotlivých řádků je možnost využít další digitální vstup pro určení začátku rámce. To lze použít například pro spuštění až je výrobek na pásu. [6]

Rozhodl jsem se pro volnoběžný režim snímání s předem nastavenou velikostí obrazu a délkou expozice. Tyto parametry je možnost dodatečně měnit v textových souborech před spuštěním programu. Kamery uloží vždy pouze jeden snímek do svého vnitřního bufferu, jedná se vždy o právě pořízený snímek a je vymazán a přepsán dalším cyklem snímkování.

Zvažoval jsem i externí synchronizaci, jelikož ve výrobě je použito krokových motorů. Výstup z PLC by tak mohl přivádět logickou úroveň na vstup kamery a ta by spustila spoušť pro zachycení snímku. Toto řešení ale bylo zavrhnuto programátorem PLC a bylo i zbytečné, protože mi nejde o celkové sestavní obrazu povrchu trubky.

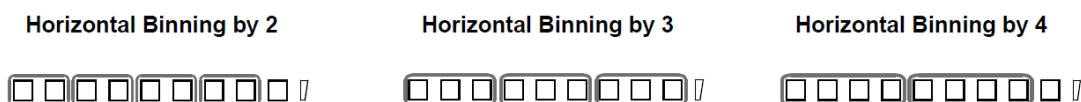
Konkrétní příkazy pro kamery a implementaci do kódu popisují v kapitole 5.

---

<sup>13</sup> Frame - rámec

### 4.5.1 Binning – slučování sousedních pixelů

Jedná se o jedno z nastavení kamery možné využít při špatných osvětlovacích podmínkách. Kamera vždy sloučí dva sousední pixely a chová se k nim jako k jednomu pixelu s dvojnásobnou plochou. Výsledek je zdvojnásobení světelné citlivosti a dvojnásobné zvýšení frekvence snímání ale rovněž se sníží rozlišení kamery na polovinu. Zorné pole zůstává stejné. [6]



Obr. 4-2: Binning [4]

## 4.6 Způsob detekce svaru

### 4.6.1 Teoretický návrh

Povrch trubky je kovově lesklý včetně svaru. To výrazně zhoršuje detekci svaru kvůli odleskům od zdrojů světla, které zhoršují viditelnost svaru i pouhým okem. Pro detekci tak bude stěžejní intenzita odraženého světla od svaru, která se liší o 50 % oproti „chromovému“ povrchu trubky mimo svar, což je v podstatě kontrast.

Z matematického hlediska budu v programu hledat lokální minima diskrétní funkce  $F(x, y)$ , kde  $x \in \langle 0; 2047 \rangle$ ,  $x \in \mathbb{Z}^{14}$ , bude představovat jednotlivé pixely a  $y \in \langle 0; 255 \rangle$ ,  $y \in \mathbb{Z}$ , bitovou hloubku jednotlivých pixelů (jejich jas). Pokud program rozezná, že v řádku převažuje počet lokálních minim, vyhodnotí je jako svar. Program bude hledat lokální minima pomocí srovnávání s průměrnou hodnotou svaru a zároveň s hodnotou chromového povrchu. Tyto hodnoty získám ze série měření pomocí kamery.

Hledání svaru započne, při založení trubky. To je zajištěno signálem od nadřazeného systému (PLC ovládající linku). Po nalezení svaru se sepne příslušné tranzistor ovládaný GPIO portem, který tím dává signál nadřazenému systému, jenž poté natočí pomocí krokových motorů danou trubku do výchozí polohy pro odejmutí pomocí dvouosého manipulátoru.

<sup>14</sup>  $\mathbb{Z}$  – Množina celých reálných čísel

## 4.6.2 Implementace detekce do kódu

Nalezení svaru zajišťuje funkce `bool hledaniSvaruH(...)`, funkce vrátí booleovskou<sup>15</sup> hodnotu `true`<sup>16</sup> nebo `false`<sup>17</sup> podle výsledku hledání. Parametry funkce jsou především, ukazatel na pole reprezentující obraz, jeho velikost a dvě prahové hodnoty normálního povrchu polotovaru načteny z textového souboru.

Identifikace svaru probíhá nejdříve zjištěním maxima a minima obrazu, poté se tyto hodnoty porovnají s prahovými hodnotami ze souboru. Pokud jsou tyto hodnoty menší než 55 % vůči prahovým hodnotám, pokračuje program do dalšího kroku identifikace.

V dalším kroku cyklus prochází v obrazu každou patnáctou hodnotu, kde u každé hodnoty kontroluje její sousední čtyři hodnoty. Pokud se vyskytne lokální extrém v podobě větší hodnoty než 5 % od průměru (lokální maximum), není hodnota v tomto kroku připočtena do bodu potenciálně nacházejícího se na svaru. Poslední podmínka pro vyhodnocení svaru je, aby minimálně 70 % takto kontrolovaných hodnot prošlo.

Tato metoda vychází z chování svaru zjištěného měřením a pozorováním. Svar nevykazuje lokální extrémy a má spíše monotónní charakter. Naproti tomu normální povrch má lokální extrémy, především lokální maxima způsobena škrábanci oděrky, a i světlem použitým v laboratorním přípravku, které nemá difuzní plát pro rovnoměrné rozložení světla na trubku. Dalším zdrojem lokálních extrémů je i samotným povrchem a jeho textura vznikla válcováním plechu do výsledného tvaru trubky. Na obrázku 4-3 je ukázka kódu a na obrázku 4-4 je snímek svaru vůči normální povrch, který byl pořízen pomocí aplikaci pylon viewer.

---

<sup>15</sup> Booleovský typ může nabývat pouze dvou hodnot 0 nebo 1.

<sup>16</sup> True – pravda

<sup>17</sup> False – nepravda

```

for(int j = 0 ;j < aVelikost; j+= 5)
{
    if(aHodnotyObrazu0[j] > max)
        max = aHodnotyObrazu0[j];
    if(aHodnotyObrazu0[j] < min)
        min = aHodnotyObrazu0[j];
}
/***** Vypočtení procent od maximální a minimální hodnoty mimo svar *****/
iMinSvaruP = min / ((double)aMinMimoSvar / 100.0);
iMaxSvaruP = max / ((double)aMaxMimoSvar / 100.0);
iPrumer = (max + min) / 2;
//cout << max << " " << min << " " << iPrumer << endl;
/*imax = max; *imin = min; *aPrumer = iPrumer;
/***** Hledání svaru *****/
if(iMaxSvaruP < 55 && iMinSvaruP < 55 && max != aMaxMimoSvar)
{
    for(; i < aVelikost; i += 15)
    {
        if(aHodnotyObrazu0[i] <= (iPrumer + ((iPrumer/100)*5)) && aHodnotyObrazu0[i - 1] <= (iPrumer + ((
            ++iPocetMinim;
        }
    if(iPocetMinim >= ((aVelikost / 15) * 0.7))
    {
        cout << "Svar nalezen pocet shod:" << iPocetMinim << "z 136" << endl;
        //cout << "Hodnota svaru " << max << " " << min << endl;
        return true;
    }
}

```

**Obr. 4-3: Ukázka kódu hledacího algoritmu**



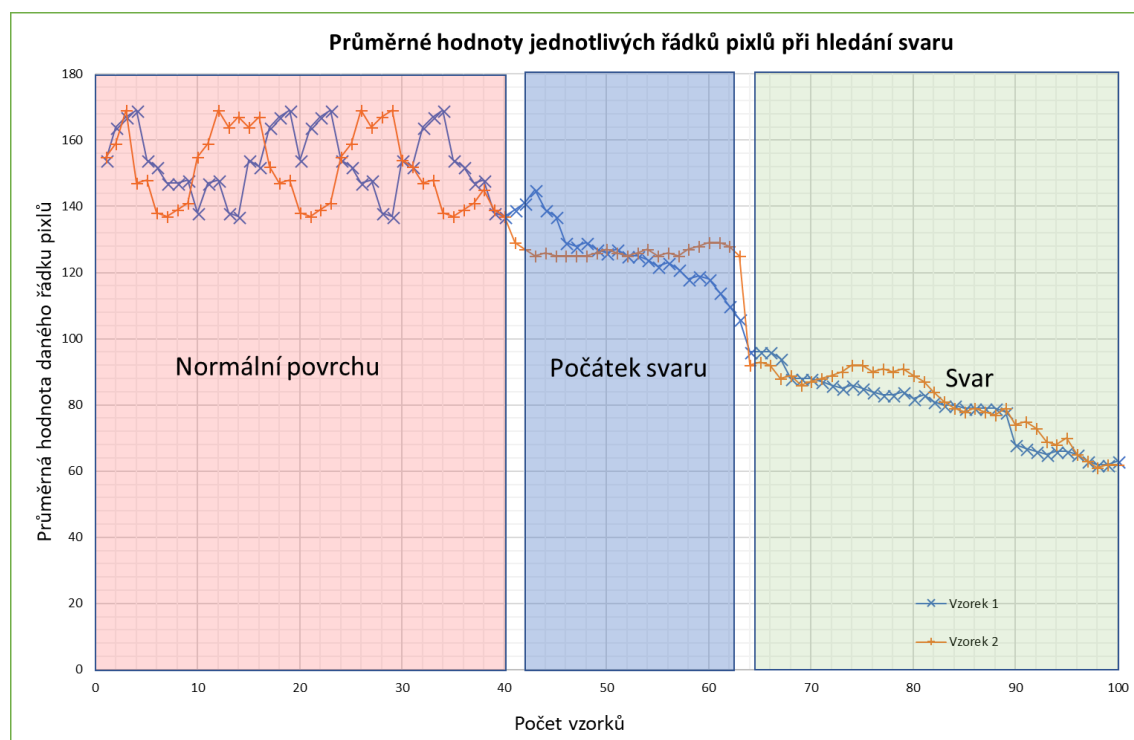
**Obr. 4-4: Svar vůči normálnímu povrchu**

### 4.6.3 Výsledky reálné detekce svaru

Hodnoty mimo svar na obrázku 4-1 jsou v maximech 254 z 255. Jedná se prakticky o přesvětlení, které jsem použil pouze pro ilustrační účely, kde je vidět lépe lokální extrémy normálního povrchu vůči malým rozdílům na svaru. V reálném použití jsem se vždy snažil dosáhnout maximální hodnoty mimo svar do 190 a maximální hodnoty svaru ne více než 90. Tyto hodnoty jsou vhodné pro detekci, ale pro prezentaci nikoliv, jelikož v tónech šedi nejsou tolik patrné lokální extrémy a svar je v podstatě kompletně tmavě šedý. Na dalším obrázku 4-5 je vykreslený graf z domácího měření pomocí Raspberry Pi s dvěma vzorky. Zde jsou maximální hodnoty mimo svar již menší a minimální hodnoty na svaru se pohybují kolem 60 z 255. Změnu intenzity jsem upravoval změnou délky expozice a digitálním zesílením kamery. Hodnota expozice činila 75  $\mu$ s a zesílení jsem nastavil 1400.

Na grafu jsou vyznačeny tři oblasti. První oblast normálního povrchu je ve skutečnosti větší, jelikož svar tvoří zhruba 8 % plochy polotovaru. Proto byl graf upraven k zobrazení pouze zájmové části. Oblast počátek svaru zachycuje první tmavé pruhy na svaru. Samotný svar je poté jasně patrný právě propadem hodnot o 50 % a útlumem lokálních extrémů.

Provedl jsem sérii několika měření, ale všechny vykazují podobnou charakteristiku. Použití právě těchto dvou měření je úmyslné, jelikož vznikly za stejných podmínek krátce po sobě a již s vyladěným algoritmem hledání svaru.



Obr. 4-5: Průměrné hodnoty intenzity ze dvou měření

## 5 IMPLEMENTACE DO RASPBERRY PI 3 B+

### 5.1 Výhody Raspberry Pi

Pro použití Raspberry Pi mě vedlo několik důvodů. V první řadě jeho kompaktnost spolu s přítomností GPIO portů. Malé rozměry Raspberry Pi umožňují snadnější montáž do rozvaděčů, kde nebývá moc místa nazbyt. Díky tomu není nutné mít separátní bednu pro PC stranou. Je ale nutné brát zřetel na to, že Raspberry nemá jako takové stínění, proto jej není vhodné instalovat do rozvaděče se silovými kabely, přenášející velké výkony, jelikož by docházelo k indukci proudů na desce, což by vedlo k případné nefunkčnosti mini počítače. Mnohé kryty pro Raspberry neumožňují montáž rozšiřující desky s výstupy a vstupy, a taky neposkytují zmíněné stínění. V tomto případě však v rozvaděči budou pouze zdroje pro giga ethernetový switch. Jedinou úpravou rozvaděče bude větrání s prachovými filtry.

Další nespornou výhodou je schopnost Raspberry Pi fungovat autonomně s přednaprogramovanou funkcionalitou. Autonomnost je podporovaná samotným operačním systémem, který je malý, rychlý, s vestavěnou možností pro ovládání Raspberry pomocí SSH protokolu nebo vzdálené plochy. V mém případě jsem použil raspbian scratch s grafickým rozhraním pro možnost ladění optické soustavy, pomocí nainstalovaného programu pylon viewer. Můj program dále umožňuje měnit parametry pro kamery pomocí textového souboru. Pokud by byla nutná kalibrace, je pro případnou obsluhu nesporně snazší měnit parametry v grafickém rozhraní.

V neposlední řadě je Raspberry vhodné do lehkého průmyslu. Často je vnímáno jenom jako výukový kit nebo vývojová deska. Existuje nespočet aplikací v průmyslu, a tím obrovské množství návodů a uživatelských zkušeností s rozličnými aplikacemi. Samotná nadace, stojící za vznikem Raspberry, jej v tomto směru podporuje.

Společnost Basler, jenž je výrobcem kamer racer, podporuje vývoj právě na embedded<sup>18</sup> platformách s procesorem ARM. U své nové řady softwaru se pyšní podporou a optimalizací pro platformy využívající procesory ARM.

Hardwarové výhody jsem již popisoval v kapitole 2.5 Raspberry Pi.

---

<sup>18</sup> Embedded – vestavěný systém se zabudovaným řídicím počítačem a potřebnými perifériemi na jedné desce.



## 5.2 Raspberry Pi a řádkové kamery

Použití řádkových kamer není běžné mimo sféru průmyslu. Proto v podstatě žádný návod ani podobnou aplikaci se mi nepodařilo najít. Zároveň podpora ARM procesorů od Basleru je relativně nová a konkrétní specifikace podporovaných systémů se objevily teprve v prosinci 2017, viz příloha Raspberry Pi dokument AW001450.

Tyto skutečnosti mě vedly k vývoji prvotní aplikace na operačním systému Windows v prostředí Visual Studia. S touto platformou se mi podařilo získat funkční kostru kódu v jazyce C++, která byla schopna rozpoznat kamery, vyčíst z nich obraz a rozeznat svar. Následující krok byl implementace do Raspberry Pi a odzkoušení téhož kódu, což proběhlo úspěšně.

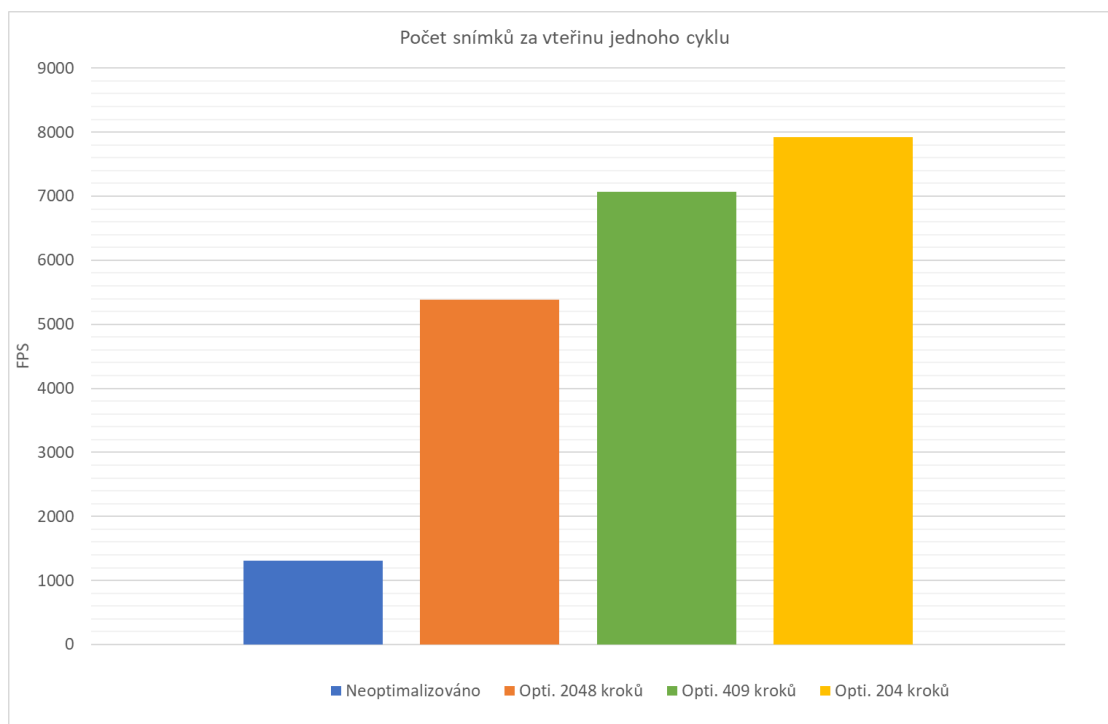
Program oproti verzi na Windows doznal změn v oblasti zahájení vyhledávání svaru a samotného cyklu hledajícího svar. Původní program po nalezení alespoň jedné kamery začal automaticky snímat a hledat svar. Vzhledem k tomu, že je vyžadována komunikace s nadřazeným systémem, tak program musí počkat na nástupní hranu signálu, aby mohl započít hledání svaru. Více o kódu je v následující kapitole 5.3.

Samotný cyklus, který hledá v datech obrazu přítomnost svaru, byl oproti verzi na Windows optimalizován. To vedlo k zvýšení rychlosti zpracování dat.

Měření délky jednoho cyklu probíhalo včetně vyzvednutí dat z obou kamer a zavolání funkce hledání svaru. Původní délka této procedury byla v průměru 766  $\mu$ s. Tato hodnota odpovídá 1304 snímkům zpracovaných za sekundu.

Po první optimalizaci, která zahrnovala odstranění matematických knihoven, a napsání vlastního cyklu pro hledání maxima, se snížil čas v průměru na 186  $\mu$ s. Z toho vychází 5385 snímků zpracovaných za sekundu. Tento nárůst je v celku markantní, ale pořád se jednalo o cyklus, který kvůli hledání maxima a minima procházel každou hodnotu.

Z předchozích testů a měření vyplynulo, že se sice vyskytují lokální extrémy především u maxima, ale nejsou pro detekci svaru až tak zásadní kvůli jejich malé četnosti. Jinak povrch mimo svar vykazoval rovnoměrné rozložení minim a maxim, viz. Obr. 4-5. To mě vedlo k většímu kroku cyklu, což na jednu stranu vede k úbytku informací o daném snímaném objektu, ale neprojevuje se to funkčností programu, z hlediska nalezení svaru. Ve verzi cyklu, kde se pohybuje krokem o velikosti deset, se tak projde pouze 204 hodnot z obrazu za účelem vyhledání maxima a minima. Tento cyklus proběhne v průměru za 126  $\mu$ s. Tato hodnota odpovídá 7917 zpracovaných snímků za sekundu. Jednotlivé verze pro krokování a jejich čas zpracování je vidět na obrázku 4-6 a v příloze číslo 5 (Měření).



**Obr. 5-1: Počet snímků zpracovaných za jednu sekundu**

Hodnoty v obrázku 4-5 a 5-1 byli vyjmuty z textových souborů vytvořených samotným programem a jsou vybrány z něco kolem 10 000 hodnot pořízených za jednu fázi vyhledávání.

### 5.3 Popis kódu a funkcionality

Kód jako takový je rozdělený do tří souborů. Jeden hlavičkový a dva zdrojové.

Hlavní kód je v zdrojovém souboru s názvem source.cpp. Začíná inicializací ovladačů pro kamery a GPIO porty. Pokračuje založením lokálních proměnných použitých v programu a inicializací tříd pro ovládání kamer (pojmenovaných jako kameraH a kameraD), otevřením a vyčtením textových souborů s nastaveními pro kamery.

Dalším krokem je přiřazení kamer nalezených na síti k inicializovaným třídám, obrázek 5-2. Kamery jsou identifikovány pomocí jejich User Define Name<sup>19</sup>, které je definováno pomocí síťové aplikace od společnosti Basler. K identifikaci je možné použít vícero metod včetně sériových čísel. Pokud kamery nejsou nalezeny, je vyhozena výjimka s chybovou hláškou a program končí. V operačním systému potom chybové hlášky odchyťávám a ukládám je do log<sup>20</sup> souboru pomocí příkazu `$ make run > stdout.txt 2> stderr.txt`.

<sup>19</sup> User Define Name – uživatel definované jméno

<sup>20</sup> Log – soubor pro záznam nějaké činnosti, v mém případě pro chyby programu.

```

//Definice nazvu kamer
GenICam::gcstring menoH = "Horni";
GenICam::gcstring menoD = "Dolni";

/***** Prirazení dostupných zarizení *****/
for(unsigned i = 0; i < tVrstva.EnumerateDevices(zarizeni); ++i)
{
    if(menoH == zarizeni[i].GetUserDefinedName())
    {
        kameraH.Attach(tVrstva.CreateDevice(zarizeni[i]));
    }
    else if(menoD == zarizeni[i].GetUserDefinedName())
    {
        kameraD.Attach(tVrstva.CreateDevice(zarizeni[i]));
    }
    else
    {
        cerr << "Zadna instance kamery nepouzita" << endl;
        throw std::runtime_error("Pokus o prirazení zarizení do instance");
        EXIT_FAILURE;
    }
}

```

**Obr. 5-2: Přirazení kamer nalezených na síti**

```

/***** Nastavení parametru obrazu a kamery *****/
if(kameraH.IsPylonDeviceAttached())
{
    kH = true;
    kameraH.Open();
    kameraH.Height.SetValue(iRadku);
    //nastavení velikosti packetu
    kameraH.GevSCPSPacketSize.SetValue(iPacketSize);
    //nastavení velikosti vystupni fornty pro zachycení posledních snímku
    kameraH.OutputQueueSize = 1;
    kameraH.Width.SetValue(iSirka);
    kameraH.ExposureTimeAbs.SetValue(iClonaH);
    kameraH.GainSelector.SetValue(GainSelector_DigitalAll);
    kameraH.GainRaw.SetValue(iHodnotaZesileniH);
}
if(kameraD.IsPylonDeviceAttached())
{
    kD = true;
    kameraD.Open();
    kameraD.Height.SetValue(iRadku);
    //nastavení velikosti packetu
    kameraD.GevSCPSPacketSize.SetValue(iPacketSize);
    //nastavení velikosti vystupni fornty pro zachycení posledních snímku
    kameraD.OutputQueueSize = 1;
    kameraD.Width.SetValue(iSirka);
    kameraD.ExposureTimeAbs.SetValue(iClonaD);
    kameraD.GainSelector.SetValue(GainSelector_DigitalAll);
    kameraD.GainRaw.SetValue(iHodnotaZesileniD);
}

```

**Obr. 5-3: Nastavení kamer**

Na dalším obrázku 5-3 je vidět samotné nastavování parametrů kamer za předpokladu, že jsou dané kamery nalezeny a přiřazeny k třídám. Pokud dojde k ukončení programu, ať už chybou nebo z jiného důvodu, dojde k zhasnutí kontrolky u linky, která značí chod programu, viz kapitola 6.

Za nastavením se program dostane do prvního `while`<sup>21</sup> cyklu, který je podmíněn proměnnou `konec`. Proměnná `konec` čte hodnotu z tlačítka na lince pomocí funkce `digitalRead(24)`<sup>22</sup> z knihovny `wiringPi.h.`, která slouží k ovládání GPIO portu Raspberry PI.

Počátek vyhledávání závisí na vstupu z PLC, a do programu se promítne stejně, jako v předchozím případě pomocí funkce `digitalRead()`. Po obdržení toho signálu se vnoří program do dalšího `while` cyklu, kde se pošle příkaz kamerám pro zahájení snímání (Obr. 5-4). Tento druhý cyklus je podmíněný nalezením svaru u obou polotovarů.

```
if(kameraD.IsPylonDeviceAttached())
    kameraD.StartGrabbing(GrabStrategy_LatestImageOnly, GrabLoop_ProvidedByUser);
if(kameraH.IsPylonDeviceAttached())
    kameraH.StartGrabbing(GrabStrategy_LatestImageOnly, GrabLoop_ProvidedByUser);
```

Obr. 5-4: Zahájení snímání kamer

Do metody pro zahájení snímání se posílají dva příkazy, určující způsob chodu kamery. `GrabStrategy_LatestImageOnly` říká kameře, necht' ukládá do svého vnitřního bufferu pouze poslední pořízený snímek.

`GrabLoop_ProvidedByUser` říká kameře, že funkci pro vyzvednutí obrazu `RetrieveResult()`<sup>23</sup> volá uživatel v kódu. Druhá možnost by byla nechat kameru automaticky vracet každý pořízený snímek. Tento postup však není vhodný účely bakalářské práce.

```
if(kH == true)
    kameraH.RetrieveResult(100, Obraz_0, TimeoutHandling_Return);
if(kD == true)
    kameraD.RetrieveResult(100, Obraz_1, TimeoutHandling_Return);
```

Obr. 5-5: Vyzvednutí obrazů od kamer

Výsledek z kamery se testuje, jestli proběhlo všechno správně. Je to vidět na obrázku 5-6. Informaci, jestli se zdařilo vyzvednutí snímku zajišťuje objekt `Obraz_0` a `Obraz_1` třídy `CGrabResultPtr`, která obsahuje další informace o vyzvednutém obraze. Po úspěšném vyzvednutí obrazu se zkopíruje do ukazatele typu `uint8_t` a odešle se jako parametr do funkce `hledaniSvaruH()`, která zajišťuje samotné hledání svaru v datech.

---

<sup>21</sup> While – dokud, cyklus se vykonává dokud je splněná podmínka.

<sup>22</sup> `digitalRead` – čtení digitální hodnoty, tj. log 0 nebo log 1.

<sup>23</sup> `RetrieveResult` – funkce pro získání obrazu, navrát' výsledek

Samotná funkce je popsána výše v podkapitole 4.6.2 Implementace detekce do kódu a její část je na obrázku 4-3.

Stejný průběh vyzvednutí a hledání je níže v kódu i pro druhou kameru, lišící se pouze názvy proměnných a objektů.

```
if(!h_svar)
{
    if(Obraz_0->GrabSucceeded())
    {
        hodnotyObrazu0 = (uint8_t *)Obraz_0->GetBuffer();
        size_t velikost = Obraz_0->GetImageSize();
        h_svar = hledaniSvaruH(hodnotyObrazu0, velikost, iPrumerMaxMimo, iPrumerMinMimo, &imax, &imin, &iPrumer);
        //iHodnotyzKamery << imax << " " << imin << " " << " " << iPrumer << endl;
        if(h_svar)
            cout << "svar nalezen Horni kamera" << endl;
    }
    else
    {
        cerr << "Nepodarilo se dostat obraz z Horni kamery" << endl;
        throw std::runtime_error("Neziskan obraz");
    }
}
```

**Obr. 5-6: Kontrola dat obrazu a vyhodnocení přítomnosti svaru**

Na konci cyklu je podmínka kontrolující, zdali došlo k nalezení svaru u obou kamer, a poté je zastaveno další pořizování snímků a vyčkává se na nové zahájení od linky.

Pokud dojde k chybě, je zachycena pomocí funkcí `catch`<sup>24</sup> podle druhu výjimky a samotné Raspberry Pi se vypne a zhasne kontrolka běhu programu. Na konci programu se již kontrolují pouze, zda byly zavřeny správně třídy kamer a ukončeno pořizování snímků. Všechny zdrojové soubory jsou v příloze 7 (Záloha Programu Malina).

---

<sup>24</sup> Catch – (zachytit) funkce obsluhující výjimky vzniklé v průběhu programu

## 6 NÁVRH INTERFACE PRO KOMUNIKACI S VÝROBNÍ LINKOU

### 6.1 Komunikační interface

Komunikace probíhá pomocí tří signálu na 24 V logice mezi Raspberry Pi a PLC. Další dva signály jsou pro obsluhu.

První signál start hledání je výstupní signál z PLC. Na nástupnou hranu logické 1 se spustí vyhledávání svarů pomocí kamer.

Další dva signály jsou z Raspberry Pi. Přivádějí logickou 1 na vstup PLC podle nalezení svaru jednotlivými kamerami.

Poslední dva signály jsou pro obsluhu. Jeden je vstup z tlačítka, který ukončí program a vypne Raspberry Pi. Druhý je výstup z Raspberry Pi spínající led kontrolku, která indikuje připravenost programu a samotný běh programu v prvním while cyklu.

Simatic 7 - 1200	Raspberry Pi 3 B+	Význam
Výstup Q1.0	Vstup GPIO pin 24	Počátek vyhledávání
Vstup I20.1	Výstup GPIO pin 23	Nalezen svar H. kamera
Vstup I20.3	Výstup GPIO pin 22	Nalezen svar D. kamera

Tab. 6.1-1: Komunikace s linkou

Raspberry Pi 3 Model B (J8 Header)		NAME	GPIO#
		3.3 VDC Power	1
		5.0 VDC Power	2
8		GPIO 8 SDA1 (I2C)	3
9		GPIO 9 SCL1 (I2C)	4
7		GPIO 7 GPCLK0	5
		Ground	6
		GPIO 15 Tx0 (UART)	15
		GPIO 16 Rx0 (UART)	16
0		GPIO 0 PCM_CLK/PWM0	1
2		GPIO 2	Ground
3		GPIO 3	GPIO 4
		3.3 VDC Power	5
12		GPIO 12 MOSI (SPI)	Ground
13		GPIO 13 MISO (SPI)	GPIO 6
14		GPIO 14 SCLK (SPI)	GPIO 10 CE0 (SPI)
		Ground	11
30		SDA0 (I2C ID EEPROM)	SCL0 (I2C ID EEPROM)
21		GPIO 21 GPCLK1	31
22		GPIO 22 GPCLK2	Ground
23		GPIO 23 PWM1	GPIO 26 PWM0
24		GPIO 24 PCM_FS/PWM1	Ground
25		GPIO 25	GPIO 27
		Ground	GPIO 28 PCM_DIN
			GPIO 29 PCM_DOUT
			29

Obr. 6-1: GPIO sběrnice [7]

Komunikace s operátorem je minimalizovaná na nejnужnější funkcionality a umožňuje mu zapnout, vypnout nebo resetovat Raspberry Pi. Jediná informace od Raspberry je, že se program vykonává a všechno je v pořádku.

Vision systém je tak navržený záměrně. Snaží se co nejvíce přiblížit jednoduchosti původního řešení s optickými snímači. Aby nebyli nutné větší zásahy do již fungující linky.

Od toho se odvíjel samotný program ovládající kamery popsany výše.



**Obr. 6-2: Rozhraní pro operátora [10]**

Červená kontrolka je pro účely bakalářské práce nahrazena zelenou, znázorňující chod programu.

Zelené tlačítko umožňuje spuštění zapnutí Raspberry Pi s automatickým startem programu po bootu<sup>25</sup>. Tlačítko funguje na principu rozepnutí, kdy Raspberry reaguje spuštěním na nástupní hranu napájecího napětí. Tuto vlastnost lze využít u tlačítka jako RESET, který prostě odpojí Raspberry Pi od napájení.

Červené tlačítko je vypínací. Je obslužené programem samotný. Pokud dojde k jeho zmačknutí, tak program po sobě nejdříve uklidí, ukončí snímání kamer, a poté dá povel pro systém, aby se vypl.

---

<sup>25</sup> Boot – zavádění operačního systému



## 7 IMPLEMENTACE DO VÝROBY

### 7.1 Výrobní linka



Obr. 7-1: Pohled na výrobní linku jako celek

Na obrázku 7-1 je výrobní linka, na kterou je navržený vision systém, popsáný v podkapitole 7.2. Linka je vlastní výroby ve společnosti Flídr s.r.o. ve spolupráci se subdodavatelem průmyslové automatizace.

V pravém dolním rohu je vidět univerzální zásobník s podávacím mechanismem založeným na principu dopravníku.

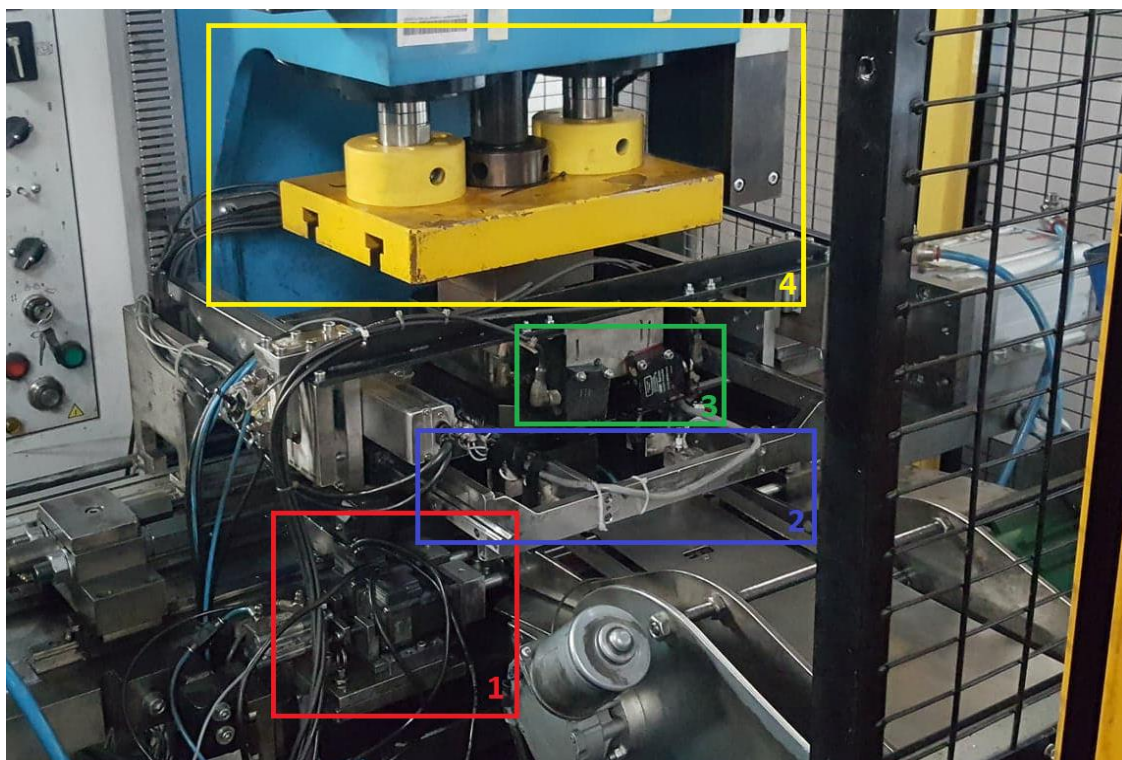
Nad ním za HMI<sup>26</sup> panelem je vidět část lisu pro lisování drážek. Na tuto čelní část budou uchyceny kamery se světly.

V levé části je potom vylisováno koncové zaoblení nožičky opěrky a plnění bedniček s hotovými kusy.

---

<sup>26</sup> HMI – Humane Machine Interface, rozhraní pro komunikaci člověka se strojem





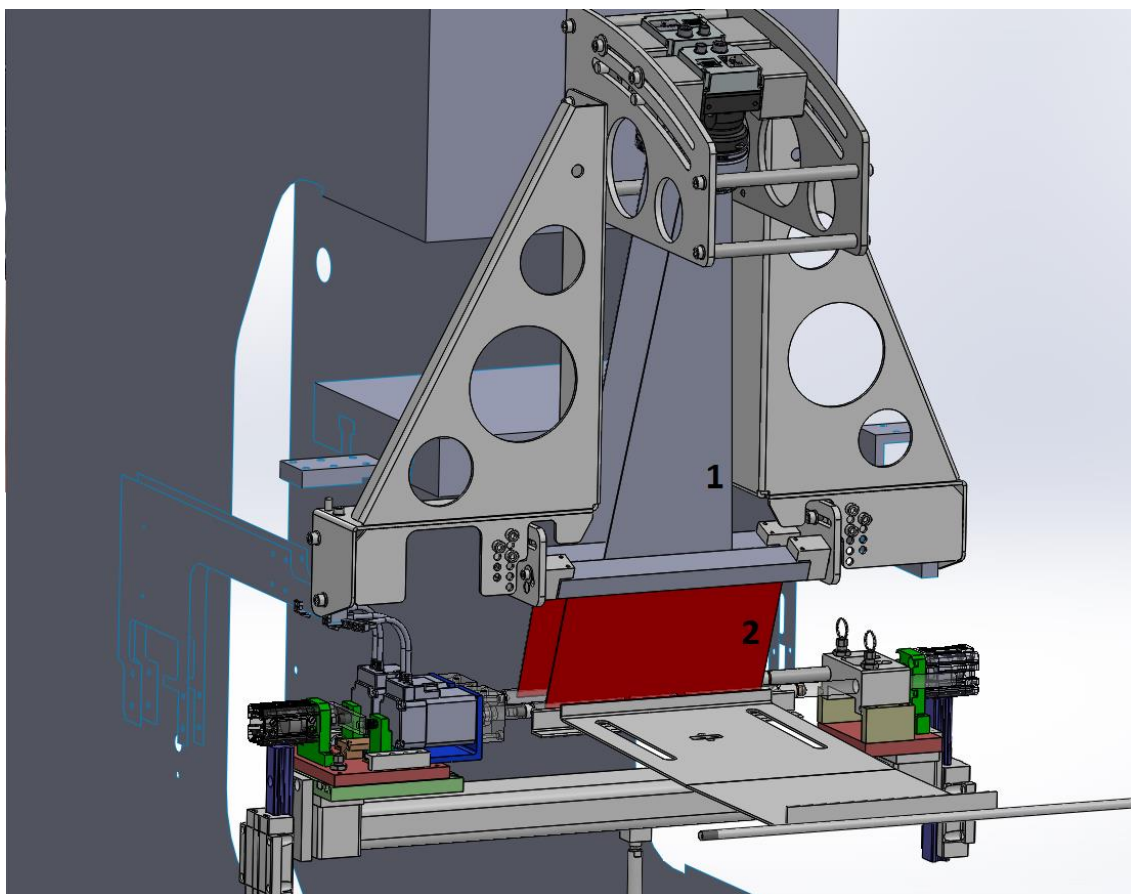
Obr. 7-2: Zájmové části linky

1. Krokové motory s trny. Při hledání svaru se nejdříve uchytí polotovár tak, že z druhé strany oproti motorům jej narazí pneumatický píst s trnem. Takto uchycený polotovár je potom vyzvednut i s motory do oblasti snímání.
2. Jednoosý lineární podavač, který zajišťuje přesun polotovaru po nalezení svaru do pozice pod lis (číslo 4).
3. Dvojce optických snímačů zajišťujících vyhledání svar pro jednu trubku. Celkem jsou tak na lince přítomny čtyři optické snímače s diskretním výstupem logické jedničky nebo nuly.
4. Hlava lisu zajišťující vylisování drážek.

## 7.2 Návrh vision systému pro linku

V této části jsou především obrázky z programu SolidWorks<sup>27</sup> pořízené konstruktérem ze společnosti Flídr s.r.o., panem Ondřejem Fialou, který vyprojektoval návrh konstrukce uchycení světla a kamer na již stávající linku. S jeho svolením tyto obrázky použiji ve své bakalářské práci k popisu vision systému pro tuto linku.

<sup>27</sup> SolidWorks – návrhový 3D CAD program



**Obr. 7-3: Návrh uchycení kamer a světel**

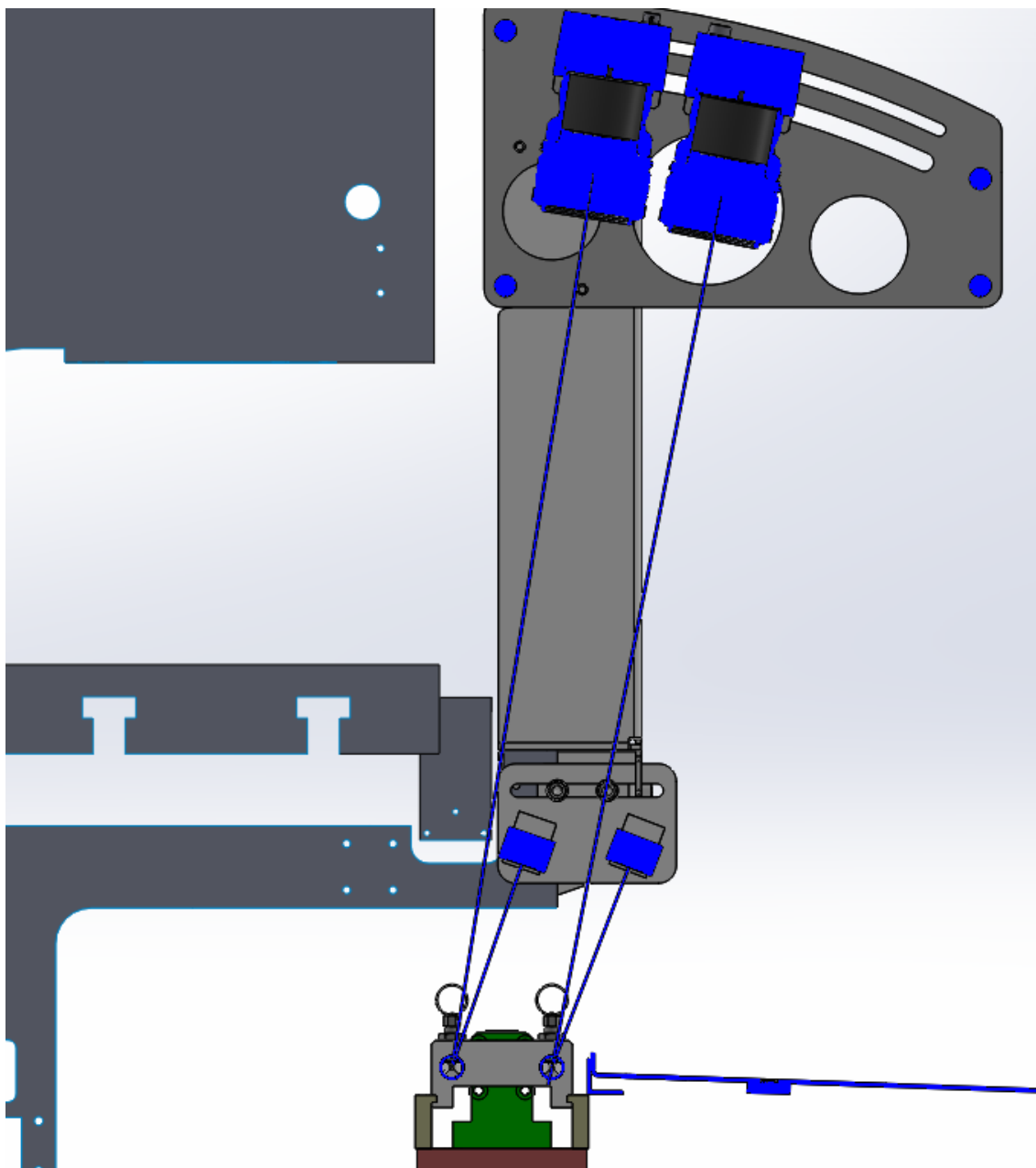
Na obrázku je 7-3 je vidět uchycení kamer a světel, společně se všemi důležitými komponenty pro bakalářskou práci, které již byly zmíněny v podkapitole 7.1 a obrázku 7-2.

1. Rovina znázorňující pohled kamery.
2. Rovina světla nasměrována tak, aby byl stejný úhel dopadu jako odrazu.

Kamery jsou upevněny v zaobleném kolejnicovém systému s poloměrem 50 cm od povrchu trubky, plus polovina výšky objektivu a celá výška f-mount<sup>28</sup> adaptéru. S pomocí malých posuvů kamery, mezi vrchní a spodní koleji, se zajišťuje možnost přesného zaměření na polotovar.

Uchycení světel je výškově nastavitelné s krokem 9 mm a možností natočení pro zlepšení odrazu do objektivu kamery. Výškový rozsah posunu kamer je od 120 mm po 93 mm od povrchu trubek.

<sup>28</sup> F-mount – druh závitu pro objektivy



**Obr. 7-4: Boční pohled na sestavu kamer a světel**

Boční pohled na sestavu znázorňuje přímý odraz od povrchu polotovaru mezi světly a kamerami.

Vzdálenost kamer přibližně 500 mm od povrchu polotovaru se bere po prostředek objektivu. Samotné kamery jsou tak umístěny výš o polovinu výšky objektivu a výšky f-mount adapteru pro kameru. vzdálenost první kamery je přesně 507 mm a druhé dolní kamery 497,92 mm. Tento rozdíl od původně zamýšlené vzdálenosti 500 mm nijak neovlivní schopnost kamer zachytit svar.

## 8 ZÁVĚR

Z měření svaru vyplynulo, že detekce svaru není až tak závislá na vlnové délce světla, pokud se pohybujeme ve viditelném spektru (380 až 750 nm), spíše jako na úhlu nasvícení a typu dopadajícího světla. Například při měření difuzního světla byla intenzita odraženého o dost menší, což výrazně zhoršuje detekci svaru. Světla podle údajů od výrobce na vzdálenost od 93 mm do 120 mm od polotovaru jsou schopny osvětlit zájmový prostor hodnotou mezi 15 000 až 17 500 luxů. Pracoviště dle měření provedeného ve společnosti je osvětleno 750 luxy. Vzhledem k těmto hodnotám je prostor detekce svaru nasvětlený konstantě, bez vlivu vnějšího rušení.

Z kapitol 2.3 a 5 je zřejmé, že řádková kamera Basler racer pro účely bakalářské práce je více než dostačující.

Použití Raspberry Pi je experiment, i když jej přímo výrobce kamery na svých stránkách uvádí jako vhodný nadřazený systém pro zpracování obrazu z kamer Basler. To také dokazují kapitoly výsledky z kapitol 2.5 a 5, kdy teoretické množství snímků přenesených přes 1000 MBit/s ethernet činí něco přes 26 000 na jednu kameru. Z kapitoly 5 praktickým testováním byly zjištěné hodnoty rychlosti cyklu kolem 7100 zpracovaných snímků dohromady z obou kamer za vteřinu, při optimalizaci na 409 kroků viz. kapitola 5.2. Na obrázku 4-5 je vidět hodnoty odrazu od polotovaru včetně svaru. Hodnoty končí v momentu, kdy program našel svar. Po správné kalibraci byl v laboratorních podmínkách svar vždy nalezen.

Laboratorní přípravek byl osazený stejnosměrným motorem. Nalezení svaru se vypsalo na konzoli. Abych byl schopen ověřit správné nalezení svaru, tak byla upravena rychlost otáčení. Po dokončení kalibrace již motor měl 1 ot/s a výpis z konzole se shodoval s místem svaru.

Algoritmus detekce svaru vycházel z požadavku výroby pro instalaci na již fungující linku s minimálními změnami a bez zásahu do programu PLC. Kvůli tomu detekce probíhá tak, že se odešle signál při první hodnotě, která se shoduje se svarem. Tato hodnota byla získaná z pěti vzorků odebraných z výroby v různých obdobích. I přes to se mi nepodařilo nalézt polotovar se svarem, který by se výrazně odlišoval od předchozích polotovarů. Tento způsob detekce není z mého pohledu ideální, pokud by se objevil kus s odlišným svarem. K tomu však podle vyjádření společnosti Flídr s.r.o. v podstatě nedochází. Svary na polotvarech mají lidským okem nepozorovatelnou toleranci v tloušťce a v současném nastavení linky tak neovlivňují detekci. Hlavní nedostatek detekce svaru z pohledu výroby je v možnosti uchycení nečistoty na svaru v místě snímání bodovými senzory intenzity. Z mého pohledu se k tomuto faktu přidává i velká plocha, dva až tři milimetry, kterou snímají senzory intenzity. Tato plocha v podstatě odpovídá tloušťce svaru na polotovaru. Tuto nevýhodu jsem se snažil odstranit

v algoritmu tím, že sčítám body s okolím, které neobsahují lokální extrémy. Pokud v daném snímku je shoda bez lokálních extrémů na alespoň 70 %, vyhodnotí se toto místo jako svar. Hodnota 70 % byla domluvena s výrobou ještě před testováním ve výrobě. Proto se může ještě změnit v průběhu zkoušení ve výrobě. Vzhledem k současné detekci pomocí senzorů intenzity by i menší číslo než 70 % vykazovalo větší přesnost. V současnosti stačí, když svar je nalezen jedním ze dvou nebo dvěma ze tří senzorů rozmístěných po délce polotovaru. Další nespornou výhodou kamer oproti senzorům je plocha snímání, která činí u kamer 150 mm na 70  $\mu$ m. Díky tomu je zajištěna lepší reakce na první hodnoty vykazující přítomnost svaru.

Pro společnost Flídr s.r.o. je použití kamer experiment. Pokud by se osvědčili i při současném způsobu detekce, byla by možnost většího zásahu do již stávajících linek nebo možnost postavit novu, která by počítala s kamerovým systémem od začátku. Ušetřilo by se tak třeba i na senzorech, které detekují správnost uchycení polotovaru a zda nevypadl z trnů. To by bez problému byly schopny určit i kamery.

Lepší způsob detekce by bylo provést otočku polotovaru alespoň o 400° a celou dobu snímat povrch. Po dokončení rotace by bylo místo svaru a jeho tloušťka přesně definovaná a provedlo by se podle toho natočení polotovaru do předem definované polohy jako teď.

Mezi další výhody Raspberry Pi se dá považovat jeho snadná dostupnost, nízká cena a možnost při poruše Raspberry vyměnit jako modul, kus za kus.

Do odevzdání bakalářské práce se bohužel nestihla reálná instalace do provozu, z několika důvodů. První důvod je odstávka linky. Datum byl stanovený nejdříve na 25. květen 2018. Druhý důvod bylo opoždění výroby dílu pro montáž na linku u subdodavatele a také opožděné dodání elektrických zdrojů a součástek pro rozvaděč.

Proto zde nemohu komentovat zlepšení spolehlivosti detekce svaru touto metodou.

Raspberry Pi se ukázalo být vhodné pro zpracování obrazu z dvou řádkových kamer. Za předpokladu, že se program přizpůsobí jeho schopnostem. Nabízí tak jistou levnou a kompaktní alternativu ke klasickým PC používaných ke zpracování obrazu.

## 9 CITOVANÁ LITERATURA

- [1] M. Sonka, V. Hlavac a R. Boyle, Image Processing, Analysis, and Machine Vision, 3rd edition, Stamford, CT 06902 USA: Cengage Learning, 2008.
- [2] T. Greš, „<http://kamery.atesystem.cz/>“, [Online]. Available: <http://kamery.atesystem.cz/know-how/line-scan-velky-pruvodce-radkovymi-kamerami/>.
- [3] atesystem. [Online]. Available: <http://kamery.atesystem.cz/produkty/podle-vyrobcce/ccs/napajeci-zdroj-pro-svetla-ccs/>.
- [4] Basler, *USER'S MANUAL FOR GigE VISION CAMERAS*, AW001183, Basler, 2017.
- [5] Basler, *How to build pylon applications on Raspberry Pi*, Basler, 2014.
- [6] elcom, „[prumyslove-kamery.cz](http://prumyslove-kamery.cz/)“, 1 10 2011. [Online]. Available: <https://www.prumyslove-kamery.cz/clanky-a-aktuality/clanky?pg=443>.
- [7] The Pi4J Project, „The Pi4J Project“, [Online]. Available: [http://pi4j.com/pins/model-3b-plus-rev1.html#Numbering\\_Scheme](http://pi4j.com/pins/model-3b-plus-rev1.html#Numbering_Scheme).
- [8] CCS, „[www.ccs-grp.com](http://www.ccs-grp.com/)“, [Online]. Available: <https://www.ccs-grp.com/products/model/1837>.
- [9] CCTV - kamerové systémy, „[http://cctv-prodejce.cz](http://cctv-prodejce.cz/)“, [Online]. Available: <http://cctv-prodejce.cz/tabulka-pro-volbu-vhodneho-objektivu>.
- [10] Schneider Electric, „AXIMA“, [Online]. Available: <https://www.axima-obchod.cz/2-tlacitka-1-signalizace-zelene-i-rude-o-ve-skrini-xal-1>. [Přístup získán 2018].

# Seznam symbolů, veličin a zkratk

FEKT	-	Fakulta elektrotechniky a komunikačních technologií
VUT	-	Vysoké učení technické v Brně
PWM	-	Pulse Width Modulation (Pulzně šířková modulace)
HMI	-	Humane Machine Interface ( Interface mezi člověkem a strojem)

# Seznam příloh

Příloha 1. Měření trubky spektrometrem – část přílohy je uložena na přiloženém CD

Příloha 2. Světlo – příloha je uložena na přiloženém CD

Příloha 3. RaspberryPi – příloha je uložena na přiloženém CD

Příloha 4. BaslerRacer – příloha je uložena na přiloženém CD

Příloha 5. Měření – část přílohy je uložena na přiloženém CD

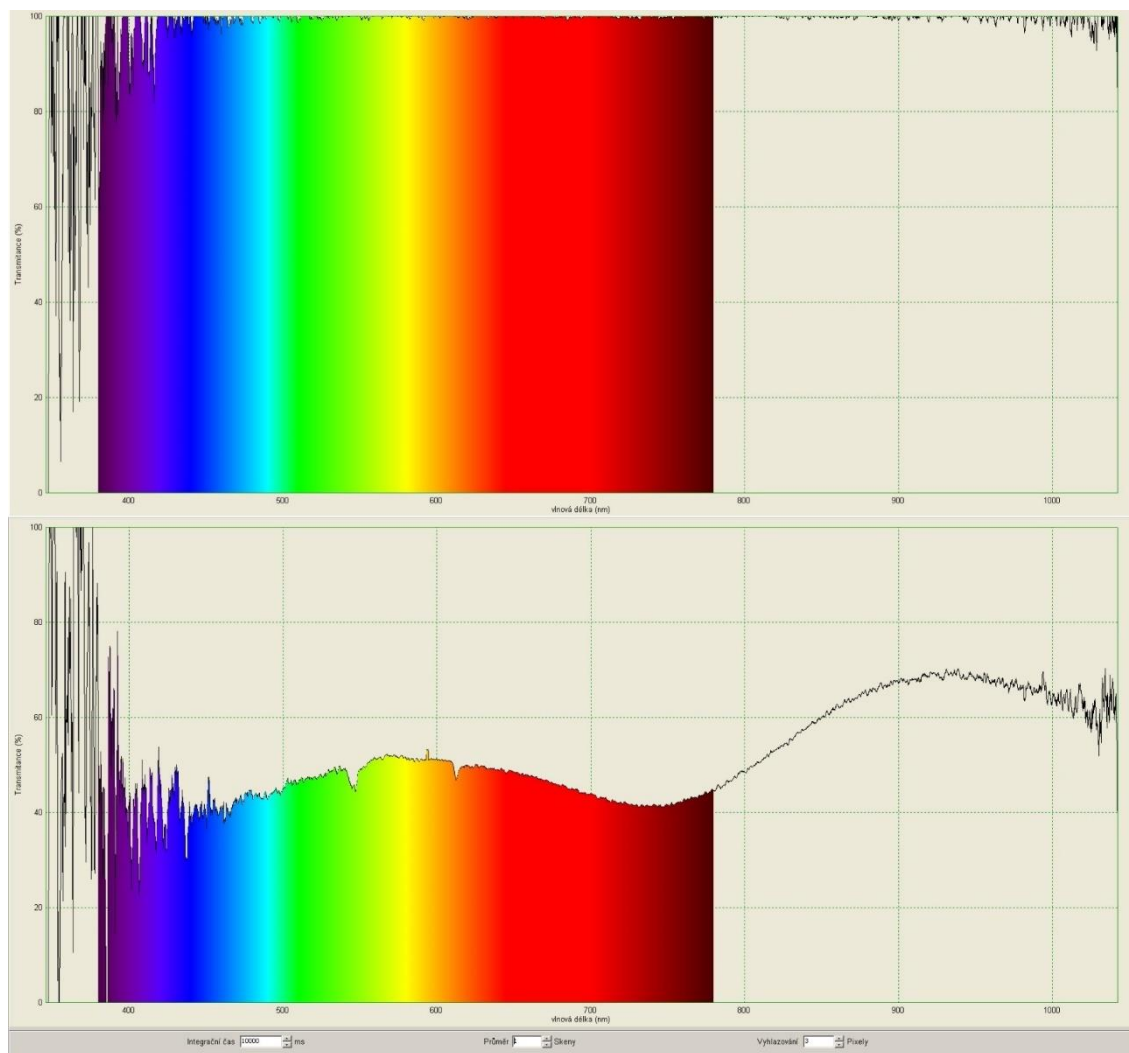
Příloha 6. Testovací pracoviště – část přílohy je uložena na přiloženém CD

Příloha 7. Záloha programu malina – příloha je uložena na přiloženém CD

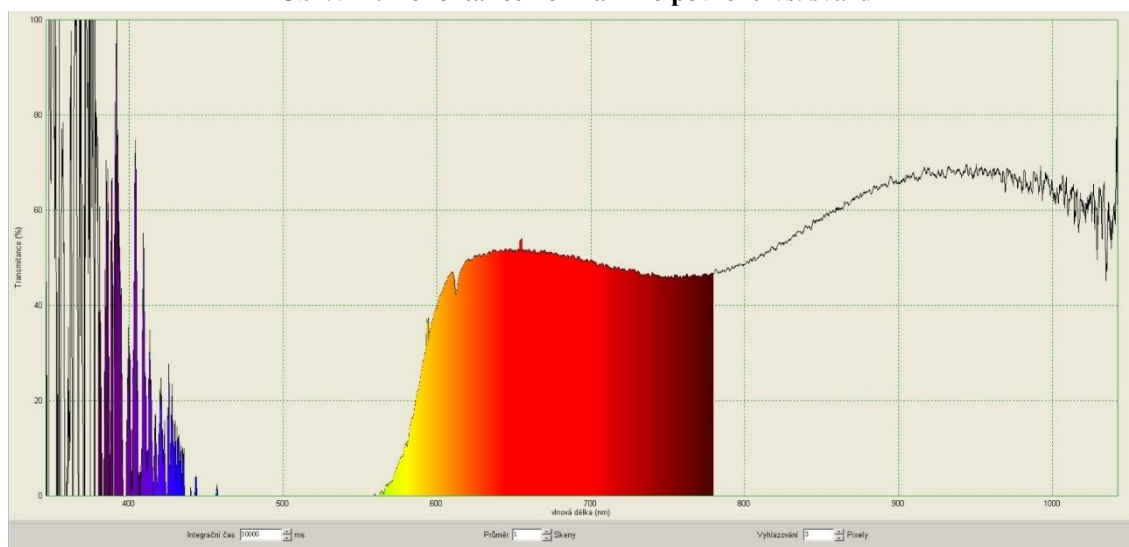
Příloha 8. Výrobní linka – část přílohy je na přiloženém CD



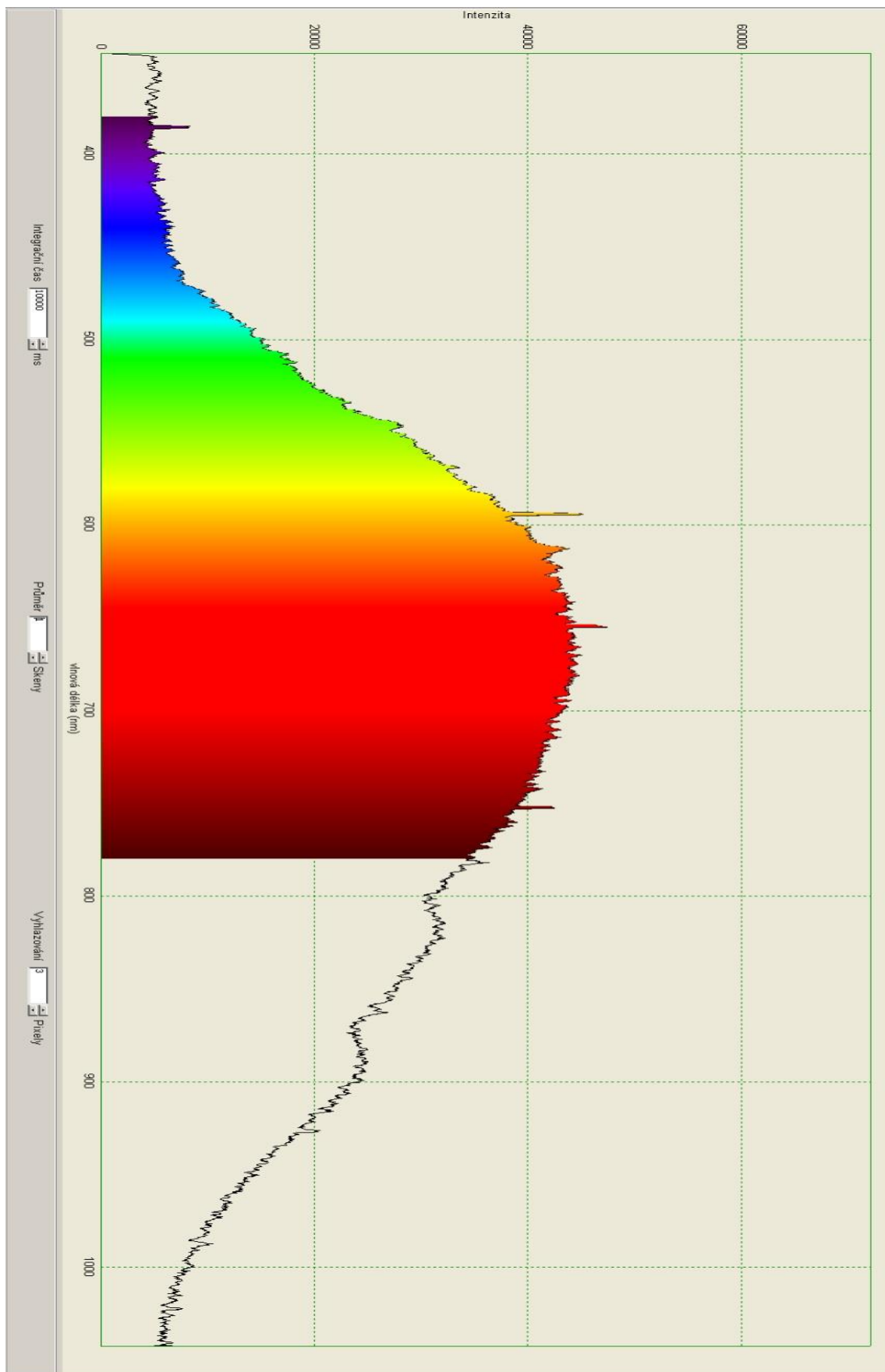
## Příloha 1 – Měření trubky spektrometrem



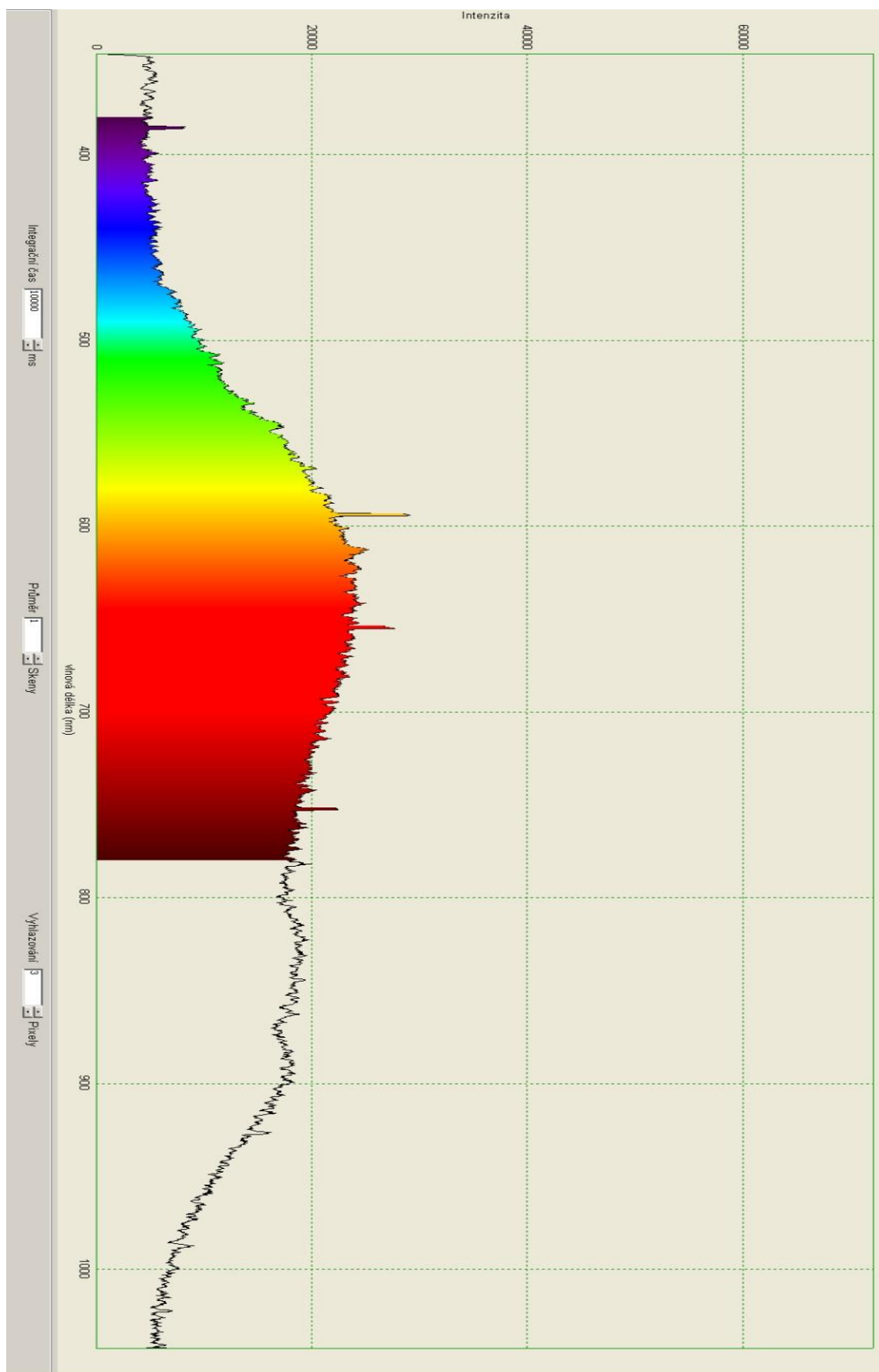
Obr. 9-1: Reflektance normálního povrchu vs. svaru



Obr. 9-2: Reflektance svaru pod červeným světlem



Obr. 9-3: Intenzita odrazu od normálního povrchu



Obr. 9-4: Intenzita odrazu od svaru

## Příloha 5 – Měření

Tabulka průměrných časů délky jednoho cyklu

	Bez optimalizace	Optim. Každý člen	Optim. Každý 5 člen	Optim. Každý 10 člen
Řádek	čas [ $\mu$ s]	čas [ $\mu$ s]	čas [ $\mu$ s]	čas [ $\mu$ s]
1	499	528	104	124
2	603	215	234	124
3	608	84	126	121
4	1196	186	58	124
5	777	165	78	123
6	567	124	135	127
7	752	144	142	121
8	591	216	137	130
9	892	99	150	126
15	792	192	138	120
16	933	189	139	135
17	564	187	144	126
18	557	186	141	122
19	1034	195	134	124
20	940	188	150	123
21	619	232	232	122
22	706	346	63	121
23	581	195	306	121
24	612	191	123	135
25	748	91	56	138
26	572	190	54	129
27	767	190	138	136
28	1073	212	142	125
29	1092	188	133	123
30	720	87	136	124
41	818	190	149	142
42	1249	125	396	132
43	777	147	68	133
44	560	148	59	128
45	508	159	142	128
46	914	239	151	125

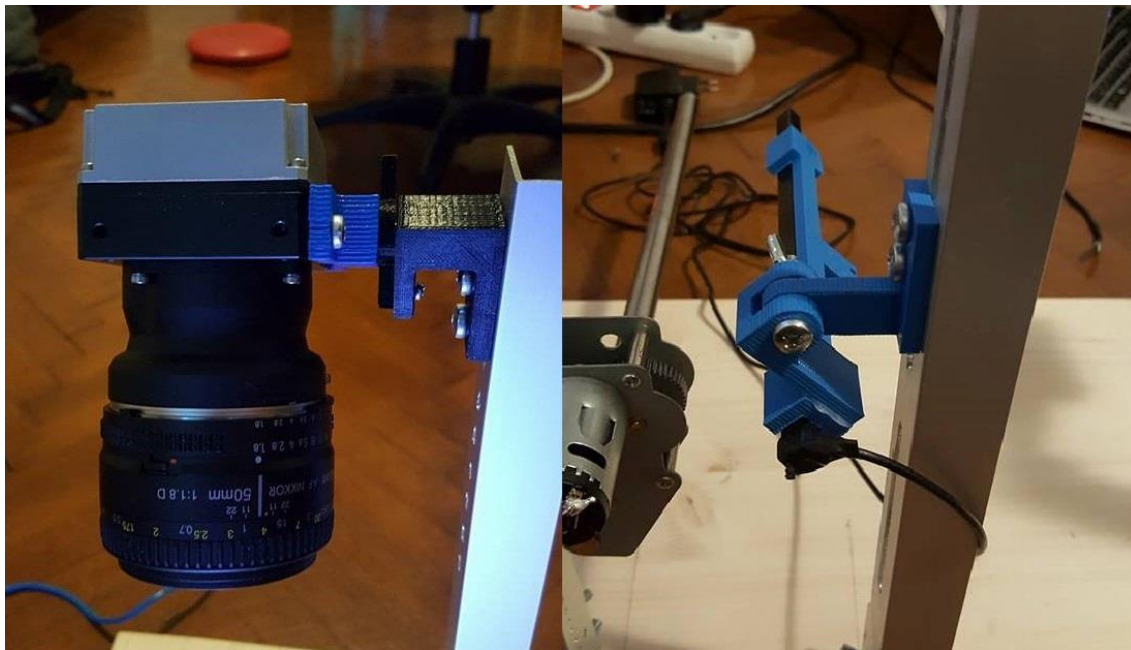
Tab. 7.2-1: Vybrané hodnoty z měření času trvání cyklu

Tabulka středních hodnot obrazu

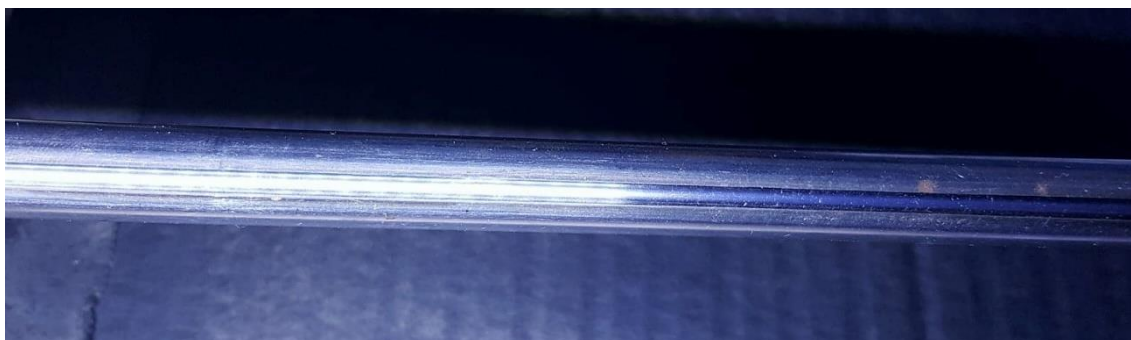
Řádek	Střední hodnota	Střední hodnota
1	164	159
2	167	169
3	169	147
4	154	148
5	152	138
6	147	137
7	147	139
8	148	141
50	128	125
51	129	126
52	127	125
53	126	125
54	127	125
55	125	125
56	125	126
57	124	127
58	122	126
59	123	125
60	121	126
61	118	127
62	119	125
63	118	126
64	114	125
65	110	127
66	106	128
87	79	78
88	79	79
89	79	78
90	79	77
91	78	79
92	68	74
93	67	75
94	66	73
95	65	69
96	66	68
97	66	70
98	65	65
99	63	63

Tab. 7.2-2: Tabulka středních hodnot obrazu

## Příloha 6 – Testovací pracoviště



Obr. 9-5: Detail kamery, světla a motoru

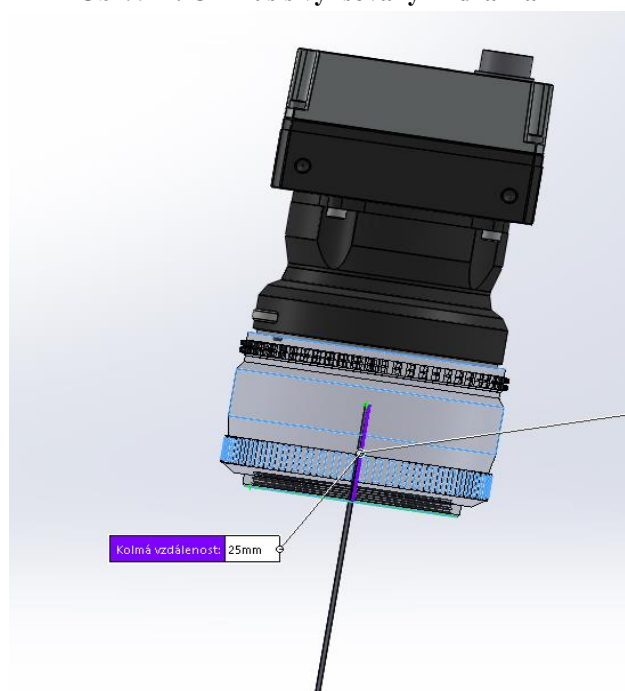


Obr. 9-6: Pohled na svar testovací pracoviště

## Příloha 8 – Výrobní linka



Obr. 9-7: OK kus s vylisovanými drážkami



Obr. 9-8: Bod odkud se měří 500 mm